Consiglio Nazionale delle Ricerche

CERIS

ISTITUTO DI RICERCA SULL'IMPRESA E LO SVILUPPO
Via Real Collegio, 30 - 10024 Moncalieri Italy.

Rapporto Tecnico N. 41
Febbraio 2012

STORAGE IN HA: MANUTENZIONE ORDINARIA E STRAORDINARIA

Giancarlo Birello, Ivano Fucile, Valter Giovanetti, Anna Perin

Via Real Collegio n. 30 – 10024 Moncalieri (TO) Italy
Codice fiscale: 80054330586 Partita IVA: 02118311006
Tel. +39 011 6824911 Fax. +39 011 6824966
http://www.ceris.cnr.it segreteria@ceris.cnr.it

Consiglio Nazionale delle Ricerche

Istituto di Ricerche sull'Impresa e Lo Sviluppo

RAPPORTO TECNICO CNR-CERIS
Anno 7, N° 41, Febbraio 2012

Direttore Responsabile
Secondo Rolfo

Direzione e Redazione
Ceris-Cnr
Istituto di Ricerca sull'Impresa e lo Sviluppo
Via Real Collegio, 30
10024 Moncalieri (Torino), Italy
Tel. +39 011 6824.911
Fax +39 011 6824.966
segreteria@ceris.cnr.it
http://www.ceris.cnr.it

Sede di Roma
Via dei Taurini, 19
00185 Roma, Italy
Tel. 06 49937810
Fax 06 49937884

Sede di Milano
Via Bassini, 15
20121 Milano, Italy
tel. 02 23699501
Fax 02 23699530

Segreteria di redazione

Enrico Viarisio
e.viarisio@ceris.cnr.it

Maria Zittino
m.zittino@ceris.cnr.it

INDICE

*STORAGE IN HA: MANUTENZIONE ORDINARIA E STRAORDINARIA**

(CLUSTER HA: ORDINARY AND EXTRAORDINARY MAINTENANCE)

Giancarlo Birello*, Ivano Fucile, Valter Giovanetti
(Cnr-Ceris, Ufficio IT)

Anna Perin
(Cnr-Ceris, Biblioteca)

Ceris-Cnr
Ufficio IT
Strada delle Cacce, 73
10135 Torino – Italy
Tel.: 011 3977533/534/535

Cnr-Ceris
Biblioteca Ceris
Via Real Collegio, 30
10024 Moncalieri (Torino), Italy
Tel.: 011 6824928

*Corresponding author: g.birello@ceris.cnr.it

ABSTRACT: This technical report analyses some ordinary and extraordinary procedures on high availability cluster discussed in Technical Report n.37, “Storage HA: active / passive cluster open-source”.
The interventions tested regarding the updating of the operating systems of the two nodes, adding space to an existing partition and adding a second disk to the DRBD replication.
The procedures have been designed considering the minimum of interruption of service, in fact most of the operations are performed on one node at a time.

KEY WORDS: cluster, open-source, linux, storage.

INTRODUZIONE

A continuazione dello sviluppo del cluster in High Availability trattato nel Rapporto Tecnico n.37, “Storage in HA: cluster attivo/passivo open-source”, sono state sperimentate alcune procedure per definire le modalità di intervento ordinarie e straordinarie sul cluster. Gli interventi sperimentati riguardano l'aggiornamento dei sistemi operativi dei due nodi, l'aggiunta di spazio ad una partizione esistente e l'aggiunta di un secondo disco alla replica DRBD.

Le procedure previste sono state studiate pensando al minimo di interruzione del servizio, in effetti la maggior parte delle operazioni sono effettuate su un nodo alla volta, permettendo così all'intero sistema di restare in produzione potendo sfruttare la disponibilità di almeno un nodo attivo. Ovviamente sono stati presi in considerazione eventuali errori e relative possibilità di ripristino della situazione precedente funzionante come la possibilità di intervento sul cluster online.

La manutenzione ordinaria è stata già utilizzata varie volte, anche con una prova di ripristino, non solo per l'aggiornamento dei sistemi operativi ma anche per l'aggiornamento dei firmware dei server che operano come nodi del cluster.

Gli interventi straordinari sono stati ripetuti varie volte per poter sperimentare e definire meglio i passi da compiere durante le operazioni di aggiunta di spazio di memorizzazione a seguito dell'incremento del numero di dischi dei nodi.

Gli esempi che riportiamo fanno stretto riferimento alla configurazione del cluster descritta nel precedente Rapporto Tecnico di cui consigliamo la lettura prima di procedere oltre.

1. MANUTENZIONE ORDINARIA

I due nodi hanno come sistema operativo linux, in particolare la distribuzione Ubuntu Server 10.04 LTS, essendo installato tutto il software dai pacchetti della distribuzione, sfruttando gli aggiornamenti di sicurezza e stabilità messi a disposizione dalla comunità, si può procedere regolarmente agli update del sistema. Questi aggiornamenti sono considerati manutenzione ordinaria del cluster e nel seguito è descritto come procedere in modo sicuro, tenendo sempre aperta la possibilità di tornare indietro nel caso una serie di aggiornamenti comprometta il funzionamento del cluster.

1.1. CONFIGURAZIONE INIZIALE

Ognuno dei due nodi è dotato di una scheda raid per la gestione delle ridondanze dei dischi a livello hardware e di 12 dischi SATA da 2 TB ciascuno. Avendo precedentemente installato lo strumento di gestione della scheda raid, dando il seguente comando possiamo vedere come sono configurati i vari dischi:

```
root@uclu1:~# hpacucli ctrl slot=1 show config

Smart Array P212 in Slot 1               (sn: xxxxxxxxxxxxxxx)

   array A (SATA, Unused Space: 0 MB)
     logicaldrive 1 (16.0 GB, RAID 5, OK)
      logicaldrive 2 (5.4 TB, RAID 5, OK)

      physicaldrive 1I:1:1 (port 1I:box 1:bay 1, SATA, 2 TB, OK)
      physicaldrive 1I:1:2 (port 1I:box 1:bay 2, SATA, 2 TB, OK)
      physicaldrive 1I:1:3 (port 1I:box 1:bay 3, SATA, 2 TB, OK)
      physicaldrive 1I:1:4 (port 1I:box 1:bay 4, SATA, 2 TB, OK)
      physicaldrive 1I:1:5 (port 1I:box 1:bay 5, SATA, 2 TB, OK, spare)

   array B (SATA, Unused Space: 0 MB)
     logicaldrive 3 (16.0 GB, RAID 5, OK)
      logicaldrive 4 (10.9 TB, RAID 5, OK)

      physicaldrive 1I:1:6 (port 1I:box 1:bay 6, SATA, 2 TB, OK)
      physicaldrive 1I:1:7 (port 1I:box 1:bay 7, SATA, 2 TB, OK)
      physicaldrive 1I:1:8 (port 1I:box 1:bay 8, SATA, 2 TB, OK)
      physicaldrive 1I:1:9 (port 1I:box 1:bay 9, SATA, 2 TB, OK)
      physicaldrive 1I:1:10 (port 1I:box 1:bay 10, SATA, 2 TB, OK)
      physicaldrive 1I:1:11 (port 1I:box 1:bay 11, SATA, 2 TB, OK)
      physicaldrive 1I:1:12 (port 1I:box 1:bay 12, SATA, 2 TB, OK)
      physicaldrive 1I:1:5 (port 1I:box 1:bay 5, SATA, 2 TB, OK, spare)

   ...
```

L'array A ha assegnati 4 dischi e l'array B ne ha assegnati 7, in entrambi i casi configurati in raid-5, mentre un 12° disco è lasciato come hot spare per entrambi gli array. Ogni array è a sua volta suddiviso in drive logici, numerati in modo crescente e univoco per l'intero controller. Abbiamo in totale 4 LD (logical drive), è possibile vedere le caratteristiche specifiche di ogni singolo LD dando il seguente comando:

```
root@uclu1:~# hpacucli ctrl slot=1 ld all show detail

Smart Array P212 in Slot 1

   array A

      Logical Drive: 1
         Size: 16.0 GB
         Fault Tolerance: RAID 5
         Heads: 255
         Sectors Per Track: 32
         Cylinders: 4112
         Strip Size: 256 KB
         Status: OK
         Array Accelerator: Enabled
         Parity Initialization Status: Initialization Completed
         Unique Identifier: xxx
         Disk Name: /dev/cciss/c0d0
         Mount Points: / 4.7 GB
         OS Status: LOCKED
         Logical Drive Label: xxx
      Logical Drive: 2
         Size: 5.4 TB
         Fault Tolerance: RAID 5
         Heads: 255
         Sectors Per Track: 32
         Cylinders: 65535
         Strip Size: 256 KB
         Status: OK
         Array Accelerator: Enabled
         Parity Initialization Status: Initialization Completed
         Unique Identifier: xxx
         Disk Name: /dev/cciss/c0d1
         Mount Points: None
         OS Status: LOCKED
         Logical Drive Label: xxx

   array B

      Logical Drive: 3
         Size: 16.0 GB
         Fault Tolerance: RAID 5
         Heads: 255
         Sectors Per Track: 32
         Cylinders: 4112
         Strip Size: 256 KB
         Status: OK
         Array Accelerator: Enabled
         Parity Initialization Status: Initialization Completed
         Unique Identifier: xxx
         Disk Name: /dev/cciss/c0d2
         Mount Points: None
         OS Status: LOCKED
         Logical Drive Label: xxx
      Logical Drive: 4
         Size: 10.9 TB
         Fault Tolerance: RAID 5
         Heads: 255
         Sectors Per Track: 32
         Cylinders: 65535
         Strip Size: 256 KB
         Status: OK
         Array Accelerator: Enabled
         Parity Initialization Status: Initialization Completed
         Unique Identifier: xxx
         Disk Name: /dev/cciss/c0d3
         Mount Points: None
         OS Status: LOCKED
         Logical Drive Label: xxx
```

Si fa notare che in quest'ultimo comando l'unità di misura della dimensione è in GiB (gibibyte) e in TiB (tebibyte) mentre nel precedente la dimensione dei dischi era in TB (terabyte).

Ad ogni LD è assegnato un device nella radice /dev/cciss del sistema operativo, abbiamo quindi due dischi virtuali da 16 GiB, c0d0 (LD1) e c0d2 (LD3), un disco da 5.4 TiB c0d1 (LD2) ed un disco da 10.9 TiB c0d3 (LD4). Il primo è usato come disco di sistema, i due più grandi per i dati ed il terzo per fare il backup della partizione di sistema prima degli interventi di manutenzione.

Per comodità i due LD da 16GiB sono stati partizionati nello stesso modo, sono state create 3 partizioni come risulta dal comando seguente:

```
root@uclu2:~# fdisk -luc /dev/cciss/c0d2

Disk /dev/cciss/c0d2: 17.2 GB, 17179607040 bytes
255 heads, 32 sectors/track, 4112 cylinders, total 33553920 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x6801fe80
           Device Boot      Start         End      Blocks   Id  System
/dev/cciss/c0d2p1            2048     9764863     4881408   83  Linux
/dev/cciss/c0d2p2        21835776    33552383     5858304   82  Linux swap /
Solaris
/dev/cciss/c0d2p3        19881984    21835775      976896   83  Linux
```

Su c0d0 (LD1) la prima partizione p1 è quella di sistema e la seconda quella di swap, mentre la terza non è utilizzata. Su c0d2 (LD3) si utilizza la seconda partizione p2 per fare il backup dell'immagine della partizione di sistema, avendo preventivamente formattato la partizione con:

```
root@uclu2:~# mkfs.ext3 /dev/cciss/c0d2p2
```

1.2. BACKUP NODO

Prima di procedere all'aggiornamento del sistema operativo di ciascun nodo, è buona norma tenere sempre i due nodi allineati rispetto la versione di software, si effettua il backup dell'intera partizione di sistema.

Partendo dalla situazione normale di funzionamento, con i due nodi attivi di cui uno master e l'altro slave, si mette il nodo slave in standby dando il seguente comando al cluster:

```
root@uclu2:~# crm node standby uclu2
```

Si inserisce a questo punto la chiavetta USB su cui è stata installata l'immagine della distribuzione SystemRescue da cui faremo avviare il nodo in standby, diamo perciò il comando seguente:

```
reboot uclu2
```

assicurandoci che il server si avvii dalla memoria USB.

Una volta entrati in SystemRescue, da riga di comando possiamo montare la partizione di backup e fare la copia dell'immagine della partizione di sistema con il tool fsarchiver, in questo modo:

```
mount /dev/cciss/c0d2p2 /mnt/backup
fsarchiver -vo savefs /mnt/backup/uclu2.fsa /dev/cciss/c0d0p1
```

una volta terminato si smonta la partizione di backup e si può riavviare il nodo, avendo tolto preventivamente la memoria USB:

```
umount /mnt/backup
reboot
```

Riavviato il nodo, che risulterà ancora in standby, si potrà mettere online dando il seguente comando al cluster:

```
crm node online uclu2
```

A questo punto conviene procedere con il backup dell'altro nodo operando esattamente allo stesso modo.

Alcune note tratte dal man del comando fsarchiver:

```
-o: overwrite the archive if it already exists instead of failing
-v: verbose mode (can be used several times to increase the level of details)
```

Per fare il restore di un filesystem di cui abbiamo l'immagine occorre dare ad esempio il seguente comando, se l'immagine contiene un solo filesystem:

```
fsarchiver restfs /mnt/backup/uclu2.fsa id=0,dest=/dev/cciss/c0d0p1
```

Per vedere il contenuto di un backup si può dare il seguente comando:

```
fsarchiver archinfo /mnt/backup/uclu2.fsa
```

1.3. AGGIORNAMENTI DI SISTEMA

Avendo a questo punto una copia di sicurezza delle partizioni di sistema dei due nodi, si può procedere con l'aggiornamento dei sistemi operativi, sempre un nodo alla volta, messo preventivamente in standby.

Per questa distribuzione si danno i seguenti comandi:

```
apt-get update
apt-get upgrade
```

Essendo comunque interdetto l'aggiornamento di release e abilitato solo l'aggiornamento dei singoli pacchetti della corrente versione di sistema operativo, in questo caso la 10.04 LTS.

2. AMPLIAMENTO PARTIZIONE ESISTENTE

Analizziamo ora il caso di un intervento straordinario di manutenzione quale l'ampliamento di una partizione dati esistente, ad esempio a seguito dell'aggiunta di nuovi dischi al sistema. Si suppone che l'ampliamento avvenga in modo identico sui due nodi, cioè ad entrambi aggiunti esattamente lo stesso numero di dischi e con le stesse caratteristiche.

2.1. STATUS INIZIALE

Si consideri il cluster in funzionamento normale, con un nodo attivo e l'altro passivo, e la partizione dati presentata al client tramite interfaccia iSCSI, client connesso e attivo, in particolare con il nodo 1 attivo ed il 2 passivo, mentre sul client la partizione dati iSCSI montata nella directory /srv/storage.

Se non esplicitamente specificato, le operazioni sono effettuate con il cluster online e il client funzionante, per maggior sicurezza sarà messo in standby solo il nodo su cui si opera.

2.2. ESPANSIONE ARRAY

Aggiunti i dischi fisicamente ai nodi, si può espandere l'array ed attribuire i dischi tramite l'utility hpacucli a riga di comando installata sui server.

Ad esempio, con il nodo 1 attivo ed il 2 in standby si daranno i comandi:

```
root@uclu2:~# hpacucli controller slot=1 array all add spares=1I:1:5
root@uclu2:~# hpacucli controller slot=1 array B add drives=1I:1:6,1I:1:7,1I:1:8
```

Quindi si metterà online il nodo 2 e dopo aver atteso la sincronizzazione DRBD, si mette in standby il nodo 1 sul quale si daranno gli stessi comandi:

```
root@uclu1:~# hpacucli controller slot=1 array all add spares=1I:1:5
root@uclu1:~# hpacucli controller slot=1 array B add drives=1I:1:6,1I:1:7,1I:1:8
```

Nell'esempio riportato, di aggiunta di 3 dischi da 2TB ad un array esistente composto da 4 dischi sempre di 2TB, occorre attendere circa 36 ore perchè terminino le operazioni di trasforming e inizializzazione dell'array espanso.

2.3. ESPANSIONE LOGICAL DRIVE

Una volta ridimensionati gli array si può procedere con i Logical Drive, sempre con la modalità precedente di operare su un nodo alla volta, messo in standby, mentre l'altro nodo rimane attivo.

Supponiamo di iniziare con il nodo 2, nell'esempio abbiamo la seguente situazione iniziale, ovviamente comune ad entrambi i nodi del cluster, dopo aver ampliato l'array ma non ancora il LD:

```
root@uclu2:~# hpacucli controller slot=1 logicaldrive 4 show detail

Smart Array P212 in Slot 1
   array B
      Logical Drive: 4
         Size: 5.9 GB
         Fault Tolerance: RAID 5
         Heads: 255
         Sectors Per Track: 32
         Cylinders: 1506
         Strip Size: 256 KB
         Status: OK
         Array Accelerator: Enabled
         Parity Initialization Status: Initialization Completed
         Unique Identifier: xxxxx
         Disk Name: /dev/cciss/c0d3
         Mount Points: None
         OS Status: LOCKED
         Logical Drive Label: xxxxx
```

A questo punto mettiamo il nodo in standby

```
root@uclu2:~# crm node standby uclu2
```

e quindi diamo il comando per espandere il Logical Drive 4, ad una determinata dimensione

```
root@uclu2:~# hpacucli controller slot=1 logicaldrive 4 modify size=8000
```

oppure occupando tutto lo spazio ancora disponibile sull'array

```
root@uclu2:~# hpacucli controller slot=1 logicaldrive 4 modify size=max
```

al warning che riceviamo possiamo rispondere yes in questo caso poiché abbiamo già verificato che non genera problemi a livello di sistema operativo o di perdita di dati

```
Warning: Extension may not be supported on certain operating systems.
         Performing extension on these operating systems can cause data to
         become inaccessible. See ACU documentation for details. Continue?
         (y/n) y
```

Anche se non necessario, abbiamo comunque atteso il termine delle operazioni prima di procedere oltre, la situazione si può rilevare tramite il comando seguente:

```
root@uclu2:~# hpacucli controller slot=1 logicaldrive 4 show detail
```

in particolare la linea riporterà

```
Parity Initialization Status: In Progress
Parity Initialization Progress: 14% complete
```

se ancora attiva l'inizializzazione, oppure

```
Parity Initialization Status: Initialization Completed
```

se l'inizializzazione terminata.

Una volta terminata l'inizializzazione abbiamo rimesso online il nodo

```
root@uclu2:~# crm node online uclu2
```

atteso la sincronizzazione del DRBD e quindi proceduto con le stesse operazioni sul nodo 1, cioè:

```
root@uclu1:~# crm node standby uclu1
root@uclu1:~# hpacucli controller slot=1 logicaldrive 4 modify size=8000
```

oppure per impegnare tutto lo spazio disponibile

```
root@uclu1:~# hpacucli controller slot=1 logicaldrive 4 modify size=max
```

Al termine abbiamo quindi rimesso online il nodo 1 col comando al cluster:

```
root@uclu1:~# crm node online uclu1
```

2.4. ESPANSIONE PARTIZIONE

Portiamo come esempio il caso di una partizione GPT (GUID Partition Table) e non MBR (Master Boot Record), essendo il caso più complesso e che riguarda in particolar modo le partizioni superiori a 2TB, non gestibili con partizioni MBR.

Essendo dedicata ad ospitare i dati, sul Logical Drive è stata creata un'unica partizione per tutto lo spazio disponibile, tramite il comando gdisk possiamo visualizzare la situazione iniziale:

```
root@uclu2:~# gdisk -l /dev/cciss/c0d3
GPT fdisk (gdisk) version 0.5.1
Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: present
Found valid GPT with protective MBR; using GPT.
Disk /dev/cciss/c0d3: 16385280 sectors, 7.8 GiB
Disk identifier (GUID): xxx
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 12288926
Total free space is 2014 sectors (1007.0 KiB)
Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048        12288926   5.9 GiB     0700  Linux/Windows data
```

Dopo l'espansione del LD potrebbe non essere immediatamente visibile il nuovo drive espanso, si può accelerare la visibilità dando il comando:

```
root@uclu2:~# hpacucli controller slot=1 show detail
```

Anche per questa operazione conviene operare sul nodo in standby, lasciando l'altro nodo attivo e quindi operativo relativo cluster e client iSCSI, sul nodo prescelto quindi diamo:

```
root@uclu2:~# crm node standby uclu2
```

e procediamo quindi con ampliare la partizione definita sul LD che è stato espanso, lo faremo tramite il comando gdisk in questo modo:

```
root@uclu2:~# gdisk /dev/cciss/c0d3

Command (? for help): x
Expert command (? for help): i
Using 1
Partition GUID code: xxx (Linux/Windows data)
Partition unique GUID: xxx
First sector: 2048 (at 1024.0 KiB)
Last sector: 12288926 (at 5.9 GiB)
Partition size: 12286879 sectors (5.9 GiB)
Attribute flags: 0000000000000000
Partition name: Linux/Windows data
```

si nota come non è ancora disponibile tutto lo spazio nuovo di cui è stato ampliato il LD, dobbiamo riposizionare la tabella alla fine del drive per poterlo utilizzare, in questo modo

```
Expert command (? for help): e
Relocating backup data structures to the end of the disk
Expert command (? for help): m
```

a questo punto se proviamo a vedere di nuovo il dettaglio del drive e relativa partizione vedremo che la dimensione del disco corrisponderà a quella nuova ampliata, mentre la partizione sarà ancora al valore originario:

```
Command (? for help): p
Disk /dev/cciss/c0d3: 16385280 sectors, 7.8 GiB
Disk identifier (GUID): 6851E4BC-1439-0309-280A-A2DFF461FD01
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 16385246
Total free space is 4098334 sectors (2.0 GiB)
Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048        12288926   5.9 GiB     0700  Linux/Windows data
```

per ampliare la partizione dovremo a questo punto rimuoverla

```
Command (? for help): d
Using 1
```

e ricrearla esattamente dallo stesso punto iniziale, ma ampliata a tutto lo spazio disponibile

```
Command (? for help): n
Partition number (1-128, default 1):
First sector (34-16385246, default = 34) or {+-}size{KMGT}: 2048
Last sector (2048-16385246, default = 16385246) or {+-}size{KMGT}:
Current type is 'Unused entry'
Hex code (L to show codes, 0 to enter raw code): 0700
Changed system type of partition to 'Linux/Windows data'
```

avremo quindi la seguente nuova situazione

```
Command (? for help): p
Disk /dev/cciss/c0d3: 16385280 sectors, 7.8 GiB
Disk identifier (GUID): xxx
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 16385246
Total free space is 2014 sectors (1007.0 KiB)
Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048        16385246   7.8 GiB     0700  Linux/Windows data

Command (? for help): i
Using 1
Partition GUID code: xxx (Linux/Windows data)
Partition unique GUID: xxx
First sector: 2048 (at 1024.0 KiB)
Last sector: 16385246 (at 7.8 GiB)
Partition size: 16383199 sectors (7.8 GiB)
Attribute flags: 0000000000000000
Partition name: Linux/Windows data
```

infine procediamo al salvataggio della nuova configurazione col seguente comando che conclude anche la sessione gdisk

```
Command (? for help): w
```

Una volta terminata la procedura si può rimettere online il nodo

```
root@uclu2:~# crm node online uclu2
```

e passare all'altro nodo per ripetere la procedura precedentemente descritta.

In breve, metteremo in standby il nodo

```
root@uclu1:~# crm node standby uclu1
```

e modificheremo la partizione tramite gdisk

```
root@uclu1:~# gdisk /dev/cciss/c0d3
```

```
Command (? for help): x
Expert command (? for help): e
Relocating backup data structures to the end of the disk
Expert command (? for help): m
Command (? for help): d
Using 1
Command (? for help): n
Partition number (1-128, default 1):
First sector (34-16385246, default = 34) or {+-}size{KMGT}: 2048
Last sector (2048-16385246, default = 16385246) or {+-}size{KMGT}:
Current type is 'Unused entry'
Hex code (L to show codes, 0 to enter raw code): 0700
Changed system type of partition to 'Linux/Windows data'
Command (? for help): w
```

A questo punto potremo quindi rimettere online il nodo

```
root@uclu1:~# crm node online uclu1
```

2.5. ESPANSIONE DRBD

L'espansione del disco DRBD è decisamente più semplice, perché effettuata in automatico dal sistema stesso, questo grazie alla configurazione adottata per il disco DRBD, i cui metadati sono di tipo flexible e memorizzati in una partizione esterna al disco (“`flexible-meta-disk /dev/cciss/c0d0p3;` ”, cfr. Rapporto Tecnico n.37, “Storage in HA: cluster attivo/passivo open-source”).

Quindi al punto precedente, quando è stato messo online il primo nodo su cui è stata estesa la partizione, avviene una prima sincronizzazione del disco DRBD ma ancora limitata alla dimensione iniziale, non essendo ancora ampliata su entrambi i nodi. Se avessimo osservato il comportamento della sincronizzazione DRBD in quel momento avremmo rilevato:

```
root@uclu2:~# cat /proc/drbd
version: 8.3.7 (api:88/proto:86-91)
GIT-hash: ea9e28dbff98e331a62bcbcc63a6135808fe2917 build by root@uclu2, 2010-11-
05 14:55:06
 0: cs:Connected ro:Secondary/Primary ds:UpToDate/UpToDate C r----
    ns:0 nr:105950 dw:105950 dr:0 al:0 bm:576 lo:0 pe:0 ua:0 ap:0 ep:1 wo:d oos:0
 1: cs:Connected ro:Secondary/Primary ds:UpToDate/UpToDate C r----
    ns:0 nr:0 dw:0 dr:0 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b oos:0
```

che la sincronizzazione avveniva quasi istantaneamente.

Viceversa, dopo aver messo online il secondo nodo su cui abbiamo esteso la partizione, a questo punto essendo entrambe le partizioni uguali e di maggior dimensione, avremmo osservato il processo di sincronizzazione estendersi su tutta la nuova dimensione e durare decisamente più tempo:

```
root@uclu1:~# cat /proc/drbd
version: 8.3.7 (api:88/proto:86-91)
GIT-hash: ea9e28dbff98e331a62bcbcc63a6135808fe2917 build by root@uclu1, 2010-11-
05 14:52:21
 0: cs:SyncTarget ro:Secondary/Primary ds:Inconsistent/UpToDate C r----
    ns:0 nr:69946 dw:69946 dr:0 al:0 bm:170 lo:1 pe:0 ua:0 ap:0 ep:1 wo:d
oos:5412
        [=================>..] sync'ed: 94.8% (5412/75172)K
        finish: 0:00:00 speed: 69,760 (69,760) K/sec
 1: cs:SyncTarget ro:Secondary/Primary ds:Inconsistent/UpToDate C r----
    ns:0 nr:757968 dw:757744 dr:0 al:0 bm:46 lo:9 pe:5498 ua:7 ap:0 ep:1 wo:b
oos:1290416
        [======>.............] sync'ed: 37.2% (1290416/2048160)K
        finish: 0:00:06 speed: 189,436 (189,436) K/sec
```

Al termine avremo la nuova situazione di sincronizzazione sull'intero disco DRBD di dimensione ampliata

```
root@uclu1:~# cat /proc/drbd
version: 8.3.7 (api:88/proto:86-91)
GIT-hash: ea9e28dbff98e331a62bcbcc63a6135808fe2917 build by root@uclu1, 2010-11-
05 14:52:21
 0: cs:Connected ro:Secondary/Primary ds:UpToDate/UpToDate C r----
    ns:0 nr:76290 dw:76290 dr:0 al:0 bm:384 lo:0 pe:0 ua:0 ap:0 ep:1 wo:d oos:0
 1: cs:Connected ro:Secondary/Primary ds:UpToDate/UpToDate C r----
    ns:0 nr:2048159 dw:2048159 dr:0 al:0 bm:126 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b
oos:0
```

Per avere un'idea dei tempi, con il nostro hardware, per sincronizzare 5.6TiB sono state necessarie circa 9 ore, mentre nell'esempio che abbiamo trattato i tempi sono stati trascurabili avendo scelto dimensioni ridotte per effettuare il test.

Una volta sincronizzati i nodi, possiamo rimetterci nella situazione normale con il nodo 1 attivo ed il nodo 2 passivo effettuando un failover in questo modo:

```
root@uclu1:~# crm node standby uclu2
root@uclu1:~# crm node online uclu2
```

2.6. ESPANSIONE SU CLIENT

Dopo aver esteso il disco DRBD possiamo procedere ad estendere la partizione agganciata via iSCSI al DRBD sul client, il serve che utilizza lo spazio per memorizzare ad esempio i contenuti di un repository.

Prima di iniziare, sul client va smontata la partizione iSCSI e prima di farlo occorre fermare le applicazioni che accedono alla partizione, ad esempio quelle che girano in tomcat come Fedora Commons.

Una volta fatto ciò, si procede con un rescan per far rilevare al sistema iSCSI la nuova configurazione del disco DRBD:

```
root@server17:~# iscsiadm --mode node -R
Rescanning session [sid: 1, target: aaa.bb.cc, portal: 1.2.3.4,3260]
```

Quindi sempre tramite gdisk, poiché anche sul client utilizziamo GPT date le dimensioni della partizione, procediamo con la estensione della partizione:

```
root@server17:~# gdisk /dev/sdb
GPT fdisk (gdisk) version 0.5.1
Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: present
Found valid GPT with protective MBR; using GPT.
Command (? for help): p
Disk /dev/sdb: 16383199 sectors, 7.8 GiB
Disk identifier (GUID): xxx
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 12286845
Total free space is 2014 sectors (1007.0 KiB)
Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048        12286845   5.9 GiB     0700  Linux/Windows data
```

viene rilevata la nuova dimensione del disco ma risulta ancora la partizione della dimensione precedente e per poter utilizzare tutto lo spazio dobbiamo riposizionare la tabella a fine disco

```
Command (? for help): x
Expert command (? for help): e
Relocating backup data structures to the end of the disk
Expert command (? for help): m
```

ora risulta infatti disponibile lo spazio fino all'ultimo settore utile

```
Command (? for help): p
Disk /dev/sdb: 16383199 sectors, 7.8 GiB
Disk identifier (GUID): F7ABD0FD-3575-1303-80B1-F6EA69B87D2B
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 16383165
Total free space is 4098334 sectors (2.0 GiB)
Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048        12286845   5.9 GiB     0700  Linux/Windows data
```

come già fatto in precedenza, per estendere la partizione dobbiamo rimuoverla e ricrearla a partire esattamente dallo stesso settore in questo modo

```
Command (? for help): d
Using 1

Command (? for help): n
Partition number (1-128, default 1):
First sector (34-16383165, default = 34) or {+-}size{KMGT}: 2048
Last sector (2048-16383165, default = 16383165) or {+-}size{KMGT}:
Current type is 'Unused entry'
Hex code (L to show codes, 0 to enter raw code): 0700
Changed system type of partition to 'Linux/Windows data'
```

risulterà quindi la nuova partizione estesa a tutto lo spazio disponibile e potremo procedere a scrivere la nuova configurazione

```
Command (? for help): p
Disk /dev/sdb: 16383199 sectors, 7.8 GiB
Disk identifier (GUID): F7ABD0FD-3575-1303-80B1-F6EA69B87D2B
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 16383165
Total free space is 2014 sectors (1007.0 KiB)
Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048        16383165   7.8 GiB     0700  Linux/Windows data

Command (? for help): w
Do you want to proceed, possibly destroying your data? (Y/N) Y
OK; writing new GPT partition table.
The operation has completed successfully.
```

Estesa la partizione resta solo da ampliare anche il filesystem in modo da poter utilizzare l'intero spazio.

Si procede prima con un check del filesystem

```
root@server17:~# e2fsck -f /dev/sdb1
e2fsck 1.41.11 (14-Mar-2010)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/sdb1: 12/384272 files (0.0% non-contiguous), 232338/1535599 blocks
```

e quindi con l'espansione

```
root@server17:~# resize2fs /dev/sdb1
resize2fs 1.41.11 (14-Mar-2010)
Resizing the filesystem on /dev/sdb1 to 2047639 (4k) blocks.
The filesystem on /dev/sdb1 is now 2047639 blocks long.
```

Infine possiamo montare di nuovo la partizione e far partire gli applicativi fermati in precedenza

```
root@server17:~# mount /dev/sdb1 /srv/storage
```

verificando la nuova dimensione disponibile al sistema

```
root@server17:~# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda1             7.5G  2.9G  4.4G  40% /
none                  243M  180K  243M   1% /dev
none                  247M     0  247M   0% /dev/shm
none                  247M   48K  247M   1% /var/run
none                  247M     0  247M   0% /var/lock
none                  247M     0  247M   0% /lib/init/rw
/dev/sdb1             7.7G  815M  6.6G  11% /srv/storage
```

A solo scopo di verifica avevamo inizialmente memorizzato sulla partizione che è stata estesa un file di una certa dimensione ed una sua copia sul filesystem di sistema del client, tramite il comando diff possiamo così confermare che i dati sulla partizione ampliata sono rimasti intatti

```
root@server17:~# diff ubuntucopy.iso /srv/storage/ubuntu.iso
```

3. AGGIUNTA NUOVA PARTIZIONE

Questo secondo caso di manutenzione straordinaria tratta l'aggiunta di un secondo disco DRBD (disk1) alla replica tra i due nodi del cluster. La configurazione di partenza è sempre la stessa, già descritta al paragrafo iniziale “Manutenzione ordinaria”, e prevede l'aggiunta di una serie di dischi (4 in questo esempio) ad entrambi i nodi che andranno a costituire un nuovo array in aggiunta ad uno già esistente e replicato via DRBD (disk0). Le dimensioni contenute delle partizioni sono dovute al carattere di test dell'esempio, senza alcuna diversità dal caso di partizioni di parecchi ordini di grandezze superiori, se non per i tempi di inizializzazione e sincronizzazione.

Si opererà in parte su entrambi i nodi e in parte solo sul nodo attivo, ma generalmente tutte le operazioni saranno effettuate con entrambi i nodi online ed il client attivo, al contrario del caso precedente, se non diversamente specificato nel testo.

3.1. NUOVO ARRAY E LOGICAL DRIVE

Dopo aver inserito fisicamente i dischi nei relativi alloggiamenti sui due server che costituiscono i due nodi del cluster, si devono definire un nuovo Array e i relativi Logical Drive in cui suddividerlo. Si effettuano le configurazioni tramite l'utility a riga di comando hpacucli, ripetendo esattamente le operazioni su entrambi i nodi. Per ovvie ragioni riporteremo una sola sequenza di comandi.

Per chiarire meglio i parametri da passare al comando, riportiamo integralmente un pezzo di man del comando hpacucli:

```
The drives parameter specifies the physical drives to be used for
   creating a logical drive on a new or existing array. If the drives specified
   are all unassigned drives, then a new array will be created with a new
   logical drive on it. If the drives specified are all assigned to an existing
   array, then a new logical drive will be created on that array.
```

Per definire un nuovo Array e assegnarli i nuovi 4 dischi occorre anche contestualmente creare un nuovo Logical Drive (LD3), che sarà di 16 GiB e in raid5, il tutto tramite il seguente comando

```
root@uclu2:~# hpacucli ctrl slot=1 create type=ld drives=1I:1:9,1I:1:10,1I:1:11,1I:1:12
raid=5 size=16000
```

Procediamo quindi con la creazione di un secondo Logical Drive (LD4) di 6 GiB in raid-5 sempre sullo stesso nuovo Array

```
root@uclu2:~# hpacucli controller slot=1 array B create type=ld size=6000 raid=5
```

A questo punto avremo la seguente situazione

```
root@uclu2:~# hpacucli ctrl slot=1 show config

Smart Array P212 in Slot 1               (sn: xxxx)
   array A (SATA, Unused Space: 0 MB)
      logicaldrive 1 (16.0 GB, RAID 5, OK)
      logicaldrive 2 (5.4 TB, RAID 5, OK)
      physicaldrive 1I:1:1 (port 1I:box 1:bay 1, SATA, 2 TB, OK)
      physicaldrive 1I:1:2 (port 1I:box 1:bay 2, SATA, 2 TB, OK)
```

```
      physicaldrive 1I:1:3 (port 1I:box 1:bay 3, SATA, 2 TB, OK)
      physicaldrive 1I:1:4 (port 1I:box 1:bay 4, SATA, 2 TB, OK)

   array B (SATA, Unused Space: 7600942 MB)
      logicaldrive 3 (16.0 GB, RAID 5, OK)
      logicaldrive 4 (5.9 GB, RAID 5, OK)
      physicaldrive 1I:1:9 (port 1I:box 1:bay 9, SATA, 2 TB, OK)
      physicaldrive 1I:1:10 (port 1I:box 1:bay 10, SATA, 2 TB, OK)
      physicaldrive 1I:1:11 (port 1I:box 1:bay 11, SATA, 2 TB, OK)
      physicaldrive 1I:1:12 (port 1I:box 1:bay 12, SATA, 2 TB, OK)
...
```

e in particolare per i nuovi LD3 e LD4

```
root@uclu1:~# hpacucli ctrl slot=1 ld all show detail

Smart Array P212 in Slot 1
   ...
    array B
      Logical Drive: 3
         Size: 16.0 GB
         Fault Tolerance: RAID 5
         Heads: 255
         Sectors Per Track: 32
         Cylinders: 4112
         Strip Size: 256 KB
         Status: OK
         Array Accelerator: Enabled
         Parity Initialization Status: Initialization Completed
         Unique Identifier: xxx
         Disk Name: /dev/cciss/c0d2
         Mount Points: None
         Logical Drive Label: xxx
      Logical Drive: 4
         Size: 5.9 GB
         Fault Tolerance: RAID 5
         Heads: 255
         Sectors Per Track: 32
         Cylinders: 1506
         Strip Size: 256 KB
         Status: OK
         Array Accelerator: Enabled
         Parity Initialization Status: Initialization Completed
         Unique Identifier: xxx
         Disk Name: /dev/cciss/c0d3
         Mount Points: None
         Logical Drive Label: xxx
```

3.2. NUOVE PARTIZIONI

Anche per partizionare i due Logical Drive si agirà esattamente in modo identico sui due nodi e riporteremo pertanto una sola serie di comandi.

Da riga di comando del nodo partizioniamo LD3 con fdisk (MBR) in tre partizioni che rispecchiano quelle di LD1, in questo modo

```
root@uclu2:~# fdisk -luc /dev/cciss/c0d2

Disk /dev/cciss/c0d2: 17.2 GB, 17179607040 bytes
255 heads, 32 sectors/track, 4112 cylinders, total 33553920 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x6801fe80

           Device Boot      Start         End      Blocks   Id  System
/dev/cciss/c0d2p1            2048     9764863     4881408   83  Linux
/dev/cciss/c0d2p2        21835776    33552383     5858304   82  Linux swap /
Solaris
/dev/cciss/c0d2p3        19881984    21835775      976896   83  Linux
```

la prima resterà inutilizzata, la seconda usata per il backup (cfr. paragrafo “Manutenzione ordinaria”) e la terza ospiterà i metadati del nuovo disco DRBD.

Per partizionare LD4 che costituirà il disco DRBD utilizziamo invece gdisk (GPT) per poter gestire dimensioni elevate, in particolare nell'esempio, prima del partizionamento abbiamo

```
root@uclu2:~# gdisk -l /dev/cciss/c0d3
GPT fdisk (gdisk) version 0.5.1
Partition table scan:
  MBR: not present
  BSD: not present
  APM: not present
  GPT: not present

Creating new GPT entries.
Disk /dev/cciss/c0d3: 12288960 sectors, 5.9 GiB
Disk identifier (GUID): xxx
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 12288926
Total free space is 12288893 sectors (5.9 GiB)
Number  Start (sector)    End (sector)  Size       Code  Name
```

e dopo aver creato la partizione, sfruttando l'intero spazio disponibile si ha

```
root@uclu2:~# gdisk -l /dev/cciss/c0d3
GPT fdisk (gdisk) version 0.5.1
Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: present
Found valid GPT with protective MBR; using GPT.
Disk /dev/cciss/c0d3: 12288960 sectors, 5.9 GiB
Disk identifier (GUID): xxx
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 12288926
Total free space is 2014 sectors (1007.0 KiB)
Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048        12288926   5.9 GiB     0700  Linux/Windows data
```

Si noti come la partizione inizia dal settore 2048, altamente raccomandato per stabilità e compatibilità GPT.

3.3. CONFIGURAZIONE DRBD DISK1

Anche per questa operazione bisogna intervenire in modo simmetrico sui due nodi, quando interessante riporteremo il comportamento di entrambi i nodi all'esecuzione dei comandi. I nodi continuano ad essere entrambi online ed il client attivo.

Innanzitutto si deve creare il file di configurazione del nuovo disk1 e in particolare avremo

```
root@uclu2:~# cat /etc/drbd.d/disk1.res
resource disk1 {
        protocol C;
       disk {
              on-io-error detach;
              fencing resource-only;
        }
        handlers {
               fence-peer "/usr/lib/drbd/crm-fence-peer.sh";
               after-resync-target "/usr/lib/drbd/crm-unfence-peer.sh";
       }
        net {
```

```
                cram-hmac-alg sha1;
                shared-secret "xxx";
        }
        syncer {
        rate 200M;
        verify-alg sha1;
        al-extents 257;
        }
        on uclu1 {
                device /dev/drbd1;
                disk /dev/cciss/c0d3p1;
                address 1.2.3.4:7790;
                flexible-meta-disk /dev/cciss/c0d2p3;
        }
        on uclu2 {
                device /dev/drbd1;
                disk /dev/cciss/c0d3p1;
                address 1.2.3.5:7790;
                flexible-meta-disk /dev/cciss/c0d2p3;
        }
}
```

Si noti in particolare la tipologia flexible dei metadati ed il loro posizionamento su una partizione esterna al disco stesso e come si debba individuare una porta (in questo caso 7790) di almeno due unità diversa da quella del disk0 (cfr. Rapporto Tecnico n.37, “Storage in HA: cluster attivo/passivo open-source”).

Su entrambi i nodi quindi faremo aggiornare la configurazione di DRBD ottenendo

```
root@uclu1:~# drbdadm adjust disk1
1: Failure: (119) No valid meta-data signature found.
       ==> Use 'drbdadm create-md res' to initialize meta-data area. <==
Command 'drbdsetup 1 disk /dev/cciss/c0d3p1 /dev/cciss/c0d2p3 flexible --set-
defaults --create-device --on-io-error=detach --fencing=resource-only' terminated
with exit code 10
```

chiaramente la non riuscita è dovuta alla mancanza dei metadati ed infatti lo status di DRBD riporterà

```
root@uclu1:~# cat /proc/drbd
version: 8.3.7 (api:88/proto:86-91)
GIT-hash: ea9e28dbff98e331a62bcbcc63a6135808fe2917 build by root@uclu1, 2010-11-
05 14:52:21
 0: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r----
    ns:677446 nr:89504 dw:766786 dr:216636 al:2728 bm:404 lo:0 pe:0 ua:0 ap:0
ep:1 wo:d oos:0
 1: cs:Unconfigured
```

Procediamo allora alla creazione dei metadati col seguente comando da ripetere su entrambi i nodi

```
root@uclu1:~# drbdadm create-md disk1
Writing meta data...
initializing activity log
NOT initialized bitmap
New drbd meta data block successfully created.
success
```

Ripetiamo quindi la procedura di aggiornamento della configurazione di DRBD, riportando nel dettaglio l'esecuzione del comando su entrambi i nodi, iniziamo dal nodo 1

```
root@uclu1:~# drbdadm adjust disk1
```

che non restituisce più errore e dallo status di DRBD si può rilevare che manca l'altro nodo ancora alla replica

```
root@uclu1:~# cat /proc/drbd
version: 8.3.7 (api:88/proto:86-91)
GIT-hash: ea9e28dbff98e331a62bcbcc63a6135808fe2917 build by root@uclu1, 2010-11-
05 14:52:21
 0: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r----
    ns:727072 nr:89504 dw:816412 dr:217076 al:2964 bm:404 lo:0 pe:0 ua:0 ap:0
ep:1 wo:d oos:0
 1: cs:WFConnection ro:Secondary/Unknown ds:Inconsistent/DUnknown C r----
    ns:0 nr:0 dw:0 dr:0 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b oos:6143440
```

Procedendo allora all'aggiornamento della configurazione sul nodo 2 otteniamo

```
root@uclu2:~# drbdadm adjust disk1
```

senza più errori e dallo status DRBD di entrambi i nodi vedremo che le repliche sono agganciate ma non ancora attivate

```
root@uclu2:/etc/drbd.d# cat /proc/drbd
version: 8.3.7 (api:88/proto:86-91)
GIT-hash: ea9e28dbff98e331a62bcbcc63a6135808fe2917 build by root@uclu2, 2010-11-
05 14:55:06
 0: cs:Connected ro:Secondary/Primary ds:UpToDate/UpToDate C r----
    ns:0 nr:727839 dw:727839 dr:0 al:0 bm:47 lo:0 pe:0 ua:0 ap:0 ep:1 wo:d oos:0
 1: cs:Connected ro:Secondary/Secondary ds:Inconsistent/Inconsistent C r----
    ns:0 nr:0 dw:0 dr:0 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b oos:6143440

root@uclu1:~# cat /proc/drbd
version: 8.3.7 (api:88/proto:86-91)
GIT-hash: ea9e28dbff98e331a62bcbcc63a6135808fe2917 build by root@uclu1, 2010-11-
05 14:52:21
 0: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r----
    ns:728182 nr:89504 dw:817522 dr:217136 al:2965 bm:404 lo:0 pe:0 ua:0 ap:0
ep:1 wo:d oos:0
 1: cs:Connected ro:Secondary/Secondary ds:Inconsistent/Inconsistent C r----
    ns:0 nr:0 dw:0 dr:0 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b oos:6143440
```

Per terminare quindi la configurazione dobbiamo scegliere quale tra i due nodi DRBD dev'essere il master, ma essendo già presente un'altra replica disk0 attiva, è ovvio che sceglieremo come master per il nuovo disk1 lo stesso nodo sul quale è master disk0, nell'esempio abbiamo disk0 master sul nodo 1 quindi daremo il comando

```
root@uclu1:~# drbdadm -- --overwrite-data-of-peer primary disk1
```

potremo osservare lo status della replica DRBD iniziale

```
root@uclu2:/etc/drbd.d# cat /proc/drbd
version: 8.3.7 (api:88/proto:86-91)
GIT-hash: ea9e28dbff98e331a62bcbcc63a6135808fe2917 build by root@uclu2, 2010-11-
05 14:55:06
 0: cs:Connected ro:Secondary/Primary ds:UpToDate/UpToDate C r----
    ns:0 nr:763253 dw:763253 dr:0 al:0 bm:47 lo:0 pe:0 ua:0 ap:0 ep:1 wo:d oos:0
 1: cs:SyncTarget ro:Secondary/Primary ds:Inconsistent/UpToDate C r----
    ns:0 nr:6041600 dw:6041600 dr:0 al:0 bm:368 lo:1 pe:640 ua:0 ap:0 ep:1 wo:b
oos:101840
       [==================>.] sync'ed: 98.4% (96/5996)M
       finish: 0:00:00 speed: 218,780 (208,328) K/sec
```

e come risulta al termine della sincronizzazione

```
root@uclu1:~# cat /proc/drbd
version: 8.3.7 (api:88/proto:86-91)
GIT-hash: ea9e28dbff98e331a62bcbcc63a6135808fe2917 build by root@uclu1, 2010-11-
05 14:52:21
 0: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r----
    ns:769919 nr:89504 dw:859260 dr:217400 al:3218 bm:404 lo:0 pe:0 ua:0 ap:0
ep:1 wo:d oos:0
 1: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r----
    ns:6143439 nr:0 dw:0 dr:6143646 al:0 bm:375 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b
oos:0
```

3.4. CONFIGURAZIONE CLUSTER

Passeremo a questo punto alla configurazione del cluster, cioè all'aggiunta del secondo disco tra le risorse gestite in HA dal cluster (cfr. Rapporto Tecnico n.37, “Storage in HA: cluster attivo/passivo open-source”). Le modifiche saranno apportate al cluster online e in particolare conviene operare sul nodo attivo, inizialmente la configurazione del cluster è la seguente

```
root@uclu1:~# crm configure show
node uclu1 \
        attributes standby="off"
node uclu2 \
        attributes standby="off"
primitive IpA ocf:heartbeat:IPaddr2 \
        params ip="1.2.3.4" cidr_netmask="24" nic="eth1" \
        op monitor interval="5s"
primitive IpArp ocf:heartbeat:SendArp \
        params ip="1.2.3.4" nic="eth1"
primitive Lvm ocf:heartbeat:LVM \
        params volgrpname="replica0"
primitive drbd-disk0 ocf:linbit:drbd \
        params drbd_resource="disk0" \
        op monitor interval="15s"
primitive iscsi lsb:iscsitarget \
        op monitor interval="15s"
primitive ping ocf:pacemaker:ping \
        params host_list="1.2.3.1" multiplier="200" dampen="5s" \
        op monitor interval="10"
primitive portblock0 ocf:heartbeat:portblock \
        params protocol="tcp" ip="1.2.3.4" portno="3260" action="block" \
        op monitor interval="10" timeout="10" depth="0"
primitive portunblock0 ocf:heartbeat:portblock \
        params protocol="tcp" ip="1.2.3.4" portno="3260" action="unblock" \
        op monitor interval="10" timeout="10" depth="0"
primitive st-uclu1 stonith:external/ipmi \
        params hostname="uclu1" ipaddr="192.168.168.168" userid="admin"
passwd="admin" interface="lan" \
        meta target-role="Started"
primitive st-uclu2 stonith:external/ipmi \
        params hostname="uclu2" ipaddr="192.168.168.169" userid="admin"
passwd="admin" interface="lan" \
        meta target-role="Started"
group HAServices portblock0 Lvm IpA IpArp iscsi portunblock0 \
        meta target-role="Started"
ms ms-drbd-disk0 drbd-disk0 \
        meta master-max="1" master-node-max="1" clone-max="2" clone-node-max="1"
notify="true"
clone pingc ping \
        meta globally-unique="false"
location HA_on_connected HAServices \
        rule $id="HA_on_connected-rule" -inf: not_defined pingd or pingd lte 0
location l-st1-uclu1 st-uclu1 -inf: uclu1
location l-st2-uclu2 st-uclu2 -inf: uclu2
colocation ms-drbd0-with-haservices inf: ms-drbd-disk0:Master HAServices
order Drbd0BLvm inf: ms-drbd-disk0:promote HAServices:start
property $id="cib-bootstrap-options" \
        dc-version="1.0.8-042548a451fce8400660f6031f4da6f0223dd5dd" \
        cluster-infrastructure="openais" \
        expected-quorum-votes="2" \
        stonith-enabled="true" \
        no-quorum-policy="ignore" \
        last-lrm-refresh="1289819733"
rsc_defaults $id="rsc-options" \
        resource-stickiness="100"
```

La configurazione del cluster è contenuta nel CIB (Cluster Information Base), un insieme di istruzioni codificate in XML, sul quale interverremo utilizzando lo strumento a riga di comando crm (Cluster Resource Manager). Ci avvaliamo della possibilità tramite crm di poter creare una copia della configurazione attuale, modificarla, verificarla e quindi farne il commit al cluster una volta controllata.

Come primo passo aggiungiamo il disk1 tra le risorse che deve gestire il cluster:

```
root@uclu1:~# crm
crm(live)# cib
crm(live)cib# new twod
INFO: 6: twod shadow CIB created
crm(twod)cib# up
crm(twod)# configure

crm(twod)configure#primitive drbd-disk1 ocf:linbit:drbd params
drbd_resource="disk1" op monitor interval="15s"
crm(twod)configure#ms ms-drbd-disk1 drbd-disk1 meta master-max="1" master-node-
max="1" clone-max="2" clone-node-max="1" notify="true"

crm(twod)configure# verify
crm(twod)configure# commit
crm(twod)configure# up
crm(twod)# cib
crm(twod)cib#commit twod
INFO: 20: commited 'twod' shadow CIB to the cluster
crm(twod)cib# use
crm(live)cib# up
crm(live)# configure
crm(live)configure# show
crm(live)configure# exit
bye
```

Quindi definiamo la collocazione del disk1 che deve coincidere con quella del disk0, cioè entrambi devono essere master sullo stesso nodo:

```
crm(live)cib# new twod --force
INFO: 6: twod shadow CIB created
crm(twod)cib# up
crm(twod)# configure

crm(twod)configure# colocation ms-drbd0-with-ms-drbd1 inf: ms-drbd-disk0:Master
ms-drbd-disk1:Master

crm(twod)configure# verify
WARNING: 15: Lvm: default timeout 20s for start is smaller than the advised 30
WARNING: 15: Lvm: default timeout 20s for stop is smaller than the advised 30
WARNING: 15: drbd-disk0: default timeout 20s for start is smaller than the
advised 240
WARNING: 15: drbd-disk0: default timeout 20s for stop is smaller than the advised
100
WARNING: 15: ping: default timeout 20s for start is smaller than the advised 60
WARNING: 15: ping: default timeout 20s for monitor_0 is smaller than the advised
60
WARNING: 15: drbd-disk1: default timeout 20s for start is smaller than the
advised 240
WARNING: 15: drbd-disk1: default timeout 20s for stop is smaller than the advised
100
ERROR: 15: cib-bootstrap-options: attribute last-lrm-refresh does not exist
```

gli avvisi ed errori sono trascurabili in questa fase e quindi procediamo

```
crm(twod)configure# commit
crm(twod)configure# up
```

possiamo osservare come tramite il comando diff si possono rilevare le modifiche rispetto la configurazione attuale attiva, per poi procedere al commit

```
crm(twod)# cib
crm(twod)cib# diff
- <cib epoch="729" num_updates="9"/>
+ <cib epoch="730" num_updates="1">
+   <configuration>
+     <constraints>
+       <rsc_colocation id="ms-drbd0-with-ms-drbd1" rsc="ms-drbd-disk0" rsc-
role="Master" score="INFINITY" with-rsc="ms-drbd-disk1" with-rsc-role="Master"
__crm_diff_marker__="added:top"/>
+     </constraints>
+   </configuration>
+ </cib>
crm(twod)cib# commit twod
INFO: 20: commited 'twod' shadow CIB to the cluster
crm(twod)cib# use
crm(live)cib# up
crm(live)# configure
crm(live)configure# show
crm(live)configure# exit
bye
```

Infine manca definire l'ordine di avvio del disco, in particolare andrà stabilito che il disco dovrà essere promosso master prima dell'avvio di tutti gli altri servizi, in questo modo:

```
crm(live)cib# new twod --force
INFO: 3: twod shadow CIB created
crm(twod)cib# diff
crm(twod)cib# up
crm(twod)# configure

crm(twod)configure# order Drbd1BLvm inf: ms-drbd-disk1:promote HAServices:start

crm(twod)configure# verify
WARNING: 9: Lvm: default timeout 20s for start is smaller than the advised 30
WARNING: 9: Lvm: default timeout 20s for stop is smaller than the advised 30
WARNING: 9: drbd-disk0: default timeout 20s for start is smaller than the advised
240
WARNING: 9: drbd-disk0: default timeout 20s for stop is smaller than the advised
100
WARNING: 9: ping: default timeout 20s for start is smaller than the advised 60
WARNING: 9: ping: default timeout 20s for monitor_0 is smaller than the advised
60
WARNING: 9: drbd-disk1: default timeout 20s for start is smaller than the advised
240
WARNING: 9: drbd-disk1: default timeout 20s for stop is smaller than the advised
100
ERROR: 9: cib-bootstrap-options: attribute last-lrm-refresh does not exist
crm(twod)configure# ptest
ptest[26896]: 2011/06/14_15:49:11 WARN: unpack_rsc_op: Processing failed op
iscsi_monitor_0 on uclu2: unknown error (1)
ptest[26896]: 2011/06/14_15:49:11 WARN: unpack_rsc_op: Processing failed op
iscsi_monitor_0 on uclu1: unknown error (1)
INFO: 10: install graphviz to see a transition graph
crm(twod)configure# commit
crm(twod)configure# up
```

fatte le verifiche ed i test, possiamo ancora una volta controllare le modifiche e quindi procedere al commit

```
crm(twod)# cib
crm(twod)cib# diff
- <cib epoch="743" num_updates="9"/>
+ <cib epoch="744" num_updates="1">
+   <configuration>
+     <constraints>
+       <rsc_order first="ms-drbd-disk1" first-action="promote" id="Drbd1BLvm"
score="INFINITY" then="HAServices" then-action="start"
__crm_diff_marker__="added:top"/>
```

```
+     </constraints>
+   </configuration>
+ </cib>
crm(twod)cib# commit twod
INFO: 15: commited 'twod' shadow CIB to the cluster
crm(twod)cib# use
crm(live)cib# up
crm(live)# configure
crm(live)configure# show
crm(live)configure# exit
bye
```

La configurazione finale del cluster risulterà quindi la seguente:

```
root@uclu1:~# crm configure show
node uclu1 \
        attributes standby="off"
node uclu2 \
        attributes standby="off"
primitive IpA ocf:heartbeat:IPaddr2 \
        params ip="1.2.3.4" cidr_netmask="24" nic="eth1" \
        op monitor interval="5s"
primitive IpArp ocf:heartbeat:SendArp \
        params ip="1.2.3.4" nic="eth1"
primitive Lvm ocf:heartbeat:LVM \
        params volgrpname="replica0"
primitive drbd-disk0 ocf:linbit:drbd \
        params drbd_resource="disk0" \
        op monitor interval="15s"
primitive drbd-disk1 ocf:linbit:drbd \
        params drbd_resource="disk1" \
        op monitor interval="15s"
primitive iscsi lsb:iscsitarget \
        op monitor interval="15s"
primitive ping ocf:pacemaker:ping \
        params host_list="1.2.3.1" multiplier="200" dampen="5s" \
        op monitor interval="10"
primitive portblock0 ocf:heartbeat:portblock \
        params protocol="tcp" ip="1.2.3.4" portno="3260" action="block" \
        op monitor interval="10" timeout="10" depth="0"
primitive portunblock0 ocf:heartbeat:portblock \
        params protocol="tcp" ip="1.2.3.4" portno="3260" action="unblock" \
        op monitor interval="10" timeout="10" depth="0"
primitive st-uclu1 stonith:external/ipmi \
        params hostname="uclu1" ipaddr="192.168.168.168" userid="admin"
passwd="admin" interface="lan" \
        meta target-role="Started"
primitive st-uclu2 stonith:external/ipmi \
        params hostname="uclu2" ipaddr="192.168.168.169" userid="admin"
passwd="admin" interface="lan" \
        meta target-role="Started"
group HAServices portblock0 Lvm IpA IpArp iscsi portunblock0 \
        meta target-role="Started"
ms ms-drbd-disk0 drbd-disk0 \
        meta master-max="1" master-node-max="1" clone-max="2" clone-node-max="1"
notify="true"
ms ms-drbd-disk1 drbd-disk1 \
        meta master-max="1" master-node-max="1" clone-max="2" clone-node-max="1"
notify="true"
clone pingc ping \
        meta globally-unique="false"
location HA_on_connected HAServices \
        rule $id="HA_on_connected-rule" -inf: not_defined pingd or pingd lte 0
location l-st1-uclu1 st-uclu1 -inf: uclu1
location l-st2-uclu2 st-uclu2 -inf: uclu2
colocation ms-drbd0-with-haservices inf: ms-drbd-disk0:Master HAServices
colocation ms-drbd0-with-ms-drbd1 inf: ms-drbd-disk0:Master ms-drbd-disk1:Master
order Drbd0BLvm inf: ms-drbd-disk0:promote HAServices:start
order Drbd1BLvm inf: ms-drbd-disk1:promote HAServices:start
property $id="cib-bootstrap-options" \
        dc-version="1.0.8-042548a451fce8400660f6031f4da6f0223dd5dd" \
        cluster-infrastructure="openais" \
        expected-quorum-votes="2" \
```

```
        stonith-enabled="true" \
        no-quorum-policy="ignore" \
        last-lrm-refresh="1289819733"
rsc_defaults $id="rsc-options" \
        resource-stickiness="100"
```

3.5. CONFIGURAZIONE ISCSI

Una volta configurato il cluster, è necessario modificare la configurazione del servizio iSCSI per rendere disponibile il nuovo disco. Verrà modificato il file di configurazione, su entrambi i nodi in modo identico e poi tramite un failover dal nodo 1 al 2 e viceversa viene fatta leggere la nuova configurazione.

Al file /etc/ietd.conf viene aggiunta la seguente parte relativa al nuovo disco:

```
Target iqn.2011-05.cnr.to:storage.disk.test.2G
        IncomingUser admin admin
        Lun 0
Path=/dev/drbd1,Type=blockio,ScsiId=CNRTOxxxxxxx,ScsiSN=xxxxxxxx,IOMode=wt
        MaxConnections         1
        InitialR2T             Yes
        ImmediateData          No
        MaxRecvDataSegmentLength 131072
        MaxXmitDataSegmentLength 131072
        MaxBurstLength         262144
        FirstBurstLength       262144
        DefaultTime2Wait       2
        DefaultTime2Retain     20
        MaxOutstandingR2T      8
        DataPDUInOrder         Yes
        DataSequenceInOrder    Yes
        ErrorRecoveryLevel     0
        HeaderDigest           None
        DataDigest             None
```

E quindi procediamo al doppio failover per far rileggere la nuova configurazione:

```
root@uclu1:~# crm node standby uclu1
root@uclu1:~# crm node online uclu1
root@uclu1:~# crm node standby uclu2
root@uclu1:~# crm node online uclu2
```

3.6. CONFIGURAZIONE CLIENT

Sul client possiamo a questo punto agganciare il nuovo disco, sempre tramite iSCSI, per far ciò procediamo prima con il discovery della nuova unità:

```
root@server17:~#iscsiadm -m discovery -t st -p 1.2.3.4
```

Quindi editare il file default auto-generato nella cartella /etc/iscsi/nodes/iqn.../1.2.3.4\,3260\,1/ per impostare l'autenticazione, cioè le linee evidenziate qui sotto:

```
# BEGIN RECORD 2.0-871
...
node.session.auth.authmethod = CHAP
node.session.auth.username = admin
node.session.auth.password = admin
...
# END RECORD
```

Sarà disponibile come disco aggiuntivo, nell'esempio abbiamo supposto come /dev/sdb, quindi potremo procedere con creare una partizione, utilizzando GPT, date le probabili dimensioni notevoli del disco, ottenendo in questo modo:

```
root@server17:~# gdisk -l /dev/sdb
GPT fdisk (gdisk) version 0.5.1
Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: present
Found valid GPT with protective MBR; using GPT.
Disk /dev/sdb: 12286879 sectors, 5.9 GiB
Disk identifier (GUID): xxx
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 12286845
Total free space is 2014 sectors (1007.0 KiB)
Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048        12286845   5.9 GiB     0700  Linux/Windows data


root@server17:~# parted -l /dev/sdb
...
Model: IET VIRTUAL-DISK (scsi)
Disk /dev/sdb: 6291MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Number  Start   End     Size    File system  Name                Flags
 1      1049kB  6291MB  6290MB  ext4         Linux/Windows data
```

Si noti come l'unità di misura della dimensione della partizione vari a seconda del comando utilizzato.

Una volta partizionato procediamo con la formattazione del file system:

```
root@server17:~# mkfs.ext4 /dev/sdb1
mke2fs 1.41.11 (14-Mar-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
384272 inodes, 1535599 blocks
76779 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=1572864000
47 block groups
32768 blocks per group, 32768 fragments per group
8176 inodes per group
Superblock backups stored on blocks:
       32768, 98304, 163840, 229376, 294912, 819200, 884736
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

Infine per poterlo utilizzare e per averlo montato automaticamente al boot effettueremo queste ultime configurazioni:

```
root@server17:~# blkid /dev/sdb1
/dev/sdb1: UUID="1234" TYPE="ext4"
```

per conoscere l'UUID della nuova partizione e quindi aggiungere la riga seguente a /etc/fstab

```
root@server17:~# printf "UUID=1234\t/srv/storage\text4\tdefaults,auto,_netdev
\t0\t0\n" >> /etc/fstab

root@server17:~# cat /etc/fstab
...
UUID=1234      /srv/storage   ext4    defaults,auto,_netdev 0       0
```

ed infine montare per la prima volta manualmente il disco

```
root@server17:~# mount /dev/sdb1 /srv/storage
```

3.7. TEST FAILOVER

Come verifica finale proviamo la copia di un file di certe dimensioni tra una partizione locale del client e la partizione ospitata sul cluster.

Se durante la copia non ci sono eventi particolari sui nodi del cluster abbiamo ottenuto:

```
root@server17:~# pv ubuntu-11.04-server-amd64.iso > /srv/storage/ubuntu.iso
 674MB 0:00:12 [  56MB/s] [==============================================>] 100%
```

Mentre abbiamo poi causato un evento di failover da un nodo all'altro durante la copia, sia da locale a remoto e viceversa, ottenendo:

```
§§failover durante copy
root@server17:~# pv ubuntu-11.04-server-amd64.iso > /srv/storage/ubuntu.iso
 674MB 0:00:31 [21.1MB/s] [==============================================>] 100%
root@server17:~# diff ubuntu-11.04-server-amd64.iso /srv/storage/ubuntu.iso
root@server17:~#

§§failover durante copy
root@server17:~# pv /srv/storage/ubuntu.iso > /root/ubuntucopy.iso
 674MB 0:00:24 [27.1MB/s] [==============================================>] 100%
root@server17:~# diff /srv/storage/ubuntu.iso ubuntucopy.iso
root@server17:~#
```

In entrambi i casi la copia ha sempre restituito nessun errore, solo il tempo è risultato allungato di circa 15 secondi, il tempo necessario al cluster per commutare di nodo durante il failover.

4. CONCLUSIONI

Le sperimentazioni effettuate sono state decisamente utili sia per poter definire meglio i passi da compiere in occasione di interventi di manutenzione e sia perché hanno richiesto in alcuni casi un approfondimento dei sistemi e degli strumenti utilizzati, ampliando la conoscenza e portando nuove idee su miglioramenti e ottimizzazioni del cluster.

Nel loro insieme, le procedure di manutenzione sono risultate semplici ed affidabili, sempre quando si operi con la dovuta attenzione e cautela, soprattutto perché si mette mano ad un sistema complesso e che viene mantenuto parzialmente attivo per eliminare disservizi per gli utenti.

Il cluster trattato in questo rapporto tecnico è ormai operativo da parecchi mesi, è la base per le partizioni di backup di utenti e gruppi ed è anche lo storage per la memorizzazione dei dati del repository del progetto Bess di digitalizzazione e conservazione di opere digitali. In questi mesi oltre agli interventi ordinari, è stato anche soggetto ad un'imprevista mancanza di energia generale, dalla quale è uscito senza alcun danno né all'hardware né ai dati contenuti, dimostrando quindi l'affidabilità complessiva su cui contavamo e per la quale è stato progettato.

Working Paper Cnr-Ceris

ISSN (*print*): 1591-0709 ISSN (*on line*): 2036-8216

Download

↓

http://www.ceris.cnr.it/index.php?option=com_content&task=section&id=4&Itemid=64

Hard copies are available on request,
please, write to:

Cnr-Ceris
Via Real Collegio, n. 30
10024 Moncalieri (Torino), Italy
Tel. +39 011 6824.911 Fax +39 011 6824.966
segreteria@ceris.cnr.it http://www.ceris.cnr.it