

## *L'Open (Source &) Ghost..*

Se c'è un argomento che suscita sempre interesse negli amministratori di sistema (e non!), a meno che non siano prossimi alla pensione, questo argomento è il backup.

Si sprecano su quest'attività le migliori risorse economiche delle varie Software House che desiderano conquistare posizioni sul mercato e le altrettante migliori risorse intellettuali dei vari System Administrator che non vogliono (o come sempre accade in realtà non possono!) ricorrere a prodotti esterni e/o a pagamento e che pertanto devono provvedere diversamente, magari dovendosi anche assumere direttamente l'onere di garantire la funzionalità della soluzione adottata/adattata al loro specifico caso.

Da questo punto di vista, una soluzione Open non può essere che la panacea di tutti i mali, se ne limitiamo il raggio d'azione ad un ragionevole ambito (che per questo articolo è tutti i sistemi x86 compatibili).

Infatti, dall'abbondante messe di informazioni, programmi e soluzioni offerte dal mondo Open Source, ho scelto di descrivere le possibilità offerte da "g4u".

Vediamo di cosa si tratta.

È un sw che risulta particolarmente indicato per la clonazione (a fini di backup, o meno..) di un intero disco (o più), anche se consente comunque il salvataggio di una singola partizione di un disco. Di qui il nome "Ghost 4 Unix"(g4u), ma che contrariamente al suo omonimo Windows:

- non soffre di antipatie per fs e partizioni diverse da quelle del S.O. nativo (e.g.: M\$\_like),
- è open (BSD Licence), liberamente distribuibile/utilizzabile per usi anche professionali!
- consente l'uso di server di rete per la dislocazione delle immagini senza le difficoltà connesse all'acquisizione di un IP via dhcp da una shell DOS (come il suo omonimo a pagamento!).

Bisogna solo disporre di una rete, un server ftp e di spazio per le immagini.

Interessante vero?

Vediamone il funzionamento nello specifico!

### Introduzione:

Il meccanismo di funzionamento è abbastanza semplice:

si fa boot da una periferica (floppy o cd) e da lì si lancia il programma di cloning/restore che altro non è che un'interfaccia verso un client ftp che trasmette al suo server i dati raccolti via dd e (più o meno leggermente) pre-compresi con gzip.

Questo potrebbe portare ad alcuni piccoli inconvenienti operativi legati per lo più ai limiti del server ftp disponibile, al fatto che non può essere effettuato (per motivi semplicemente di sicurezza!) su rete pubblica, al fatto che il sw g4u gira solo su macchine del tipo x86 compatibile e che la clonazione di un disco anche se per metà vuoto potrebbe portare via abbastanza spazio.

A questo ultimo inconveniente è stata proposta più di una soluzione e quelle adottate migliorano sempre di più.

Non resta pertanto che sperare che i driver già a bordo dell'immagine da cui facciamo boot, si adattino alla ns. scheda di rete (quelle integrate – specie sui laptop - sono sempre terribili da accontentare! Ma non disperate mai.. ;-P) ed avere sufficiente spazio per le immagini.

Ah.. ovviamente se volete clonare la singola partizione e non l'intero disco, è possibile a partire dalla versione 1.12 di g4u, così come vi ho già anticipato!

L'indirizzo dove reperire il software e fare riferimento per qualsiasi dubbio o perplessità è:

- <http://rfhs8012.fh-regensburg.de/~feyrer/g4u>

### Prerequisiti:

Non sono necessari particolari prerequisiti; potete anche non avere alcuna partizione unix installata ed usare un boot-cd od un boot-floppy per partire in modalità NetBSD (è sempre uno unix!) ed eseguire da lì o l'`upload{disk | part}` (genera l'immagine) o lo `slurp{disk | part}` (che invece la ripristina), sia per l'intero disco, che per le singole partizioni.

```
No change; not updating disk label.
sd1: Device not configured
sd2: Device not configured
sd3: Device not configured
-----
Welcome to g4u Harddisk Image Cloning V1.12!

* To upload disk-image to FTP, type:    uploaddisk serverIP [image] [disk]
* To upload partition to FTP, type:     uploadpart serverIP [image] [disk+p
* To install harddisk from FTP, type:    slurpdisk serverIP [image] [disk]
* To install partition from FTP, type:   slurppart serverIP [image] [disk+p
* To copy disks locally, type:           copydisk disk0 disk1

[disk] defaults to wd0 for first IDE disk, [disk+part] defaults to wd0d for
the whole first IDE disk. Use wd1 for second IDE disk, sd0 for first SCSI
disk, etc. Default image for slurpdisk is 'rwd0d.gz'. Run 'dmesg' to see
boot messages, 'disks' for recognized disks, 'parts <disk>' for list of
(BSD-type!) partitions on disk "<disk>" (wd0, ...), run any other commands
without args to see usage message.

Enjoy!                                     Send comments to hubert@feyrer
                                           http://www.feyrer.de/g4u/
-----
#
```

In questo ultimo caso – cioè gestione di singole partizioni – oppure per chi con Unix non ha dimestichezza, ma vorrebbe comunque usare g4u, vale la pena di spendere due parole su “come” le partizioni di un disco vengono viste in modo più o meno differente da NetBSD rispetto a linux e rispetto a windows; anzi cominciamo proprio da questo ultimo:

- Win - la confusione regna sovrana: la partizione primaria del primo disco viene vista come unità C:, dopo di che le restanti vengono individuate in una modalità abbastanza variabile.. in genere le altre partizioni primarie hanno le lettere successive, salvo la presenza di partizioni estese sullo stesso disco della prima primaria (quella del sistema operativo per intenderci!), o di una loro successiva variazione di configurazione, che potrebbe far variare qualcosa nell'ordine utilizzato!
- Linux – i dischi sono individuati rispetto alla posizione con cui si attestano sui controller ide; quindi le unità sul primary master assumono il nome di hda (primary slave hdb), mentre le unità sul secondary master assumono il nome di hdc (secondary slave hdd). Le unità scsi sono invece “letterate” da <a> a <f> in funzione della posizione che ricoprono sulla catena scsi, quindi sda la prima, sdb la seconda e così via.. Eventuali partizioni su questi dischi conservano il nome del disco ed a questo aggiungono il loro numero relativo: la prima partizione del disco sul primary master sarà la hda1, la seconda partizione sul secondary master sarà hdc2, la terza partizione sul

secondary slave sarà hdd3 e così via; lo stesso vale per le unità scsi, per le quali le eventuali partizioni vengono “numerate” progressivamente: sda1, sda2, sdb1, ..

- NetBSD – vale lo stesso discorso fatto per linux ma le unità ide vengono “chiamate” semplicemente wd#, dove anche qui vale il semplice ordine con cui vengono scoperte sui controller ide (come per lo scsi!) e numerate progressivamente [0, 1, ..] (stavolta lo 0 è incluso!), per cui la somiglianza con le unità scsi è molto più alta che per i primi due sistemi riportati! Le partizioni quindi vengono “letterate” e non “numerate” come invece avevamo visto in linux per le partizioni delle unità scsi; avremo così: wd0e, wd1f, etc... Sì, ci avrete fatto già caso, ma lo dico lo stesso: le lettere da “a” a “d” sono riservate per altri usi, per cui la “letterazione” delle vostre partizioni partirà da e, f, ...

In ogni caso se avete dubbi è sempre possibile chiedere al sistema di “mostrarvi” come vede le vs. partizioni, grazie al comando “`parts wd#`” [dove con “#” intendo ovviamente un numero intero!] che è comunque disponibile sulla console minimale messa a disposizione dal ns. *boot-media* (tipicamente floppy o cd, ma se volete farvi il vs. boot-dvd, niente in contrario!).

#### Attività:

Si fa boot da un *boot-media* (vedi sopra) e da lì (salvo preventiva attività di ricognizione dello stato delle partizioni - `parts`), si lancia il programma di cloning/restore [accertatevi che il server dhcp ci abbia assegnato l'indirizzo corretto (`ifconfig -a`, se non ricordiamo male.. ;-D)].

```
Enjoy!                                     Send comments to hubert@feyrer
                                           http://www.feyrer.de/g4u/
-----
# uploaddisk mimmo@192.168.1.210 back_nt.gz
( cat $tmpfile ; dd progress=1 if=/dev/rwd0d bs=1M ! gzip -9 ) ! ftp -n
tmpfile:
open 192.168.1.210
user mimmo
bin
put - back_nt.gz
bye
5
4
3
2
1
working...
Connected to 192.168.1.210.
dd: /dev/rwd0d: Operation not supported
220 leonardo FTP server (Version 4.1 Sat Sep 7 14:31:53 CDT 2002) ready.
331 Password required for mimmo.
Password: █
```

Questo ovviamente da per scontato che o avete già un vs. account sul server ftp o ne avete creato uno ad hoc (per chi non ha fantasia, quello di default invocato dal programma è: “*install*”).

A seconda di come sceglierete di usare il programma, potrebbe cambiare leggermente la procedura

da utilizzare:

- Clonazione di un intero hard disk: "uploaddisk your.ftp.server.com filename.gz"  
(Da per scontato che l'assenza di parametri a fine riga comando implichi il riferimento al device `rwd0d` – cioè tutto il primo disco in modalità raw!)
- Di una singola partizione: "uploadpart your.ftp.server.com filename.gz wd0e"  
(Che ha invece come input la singola partizione specificata a fine riga comando)  
[Ovviamente i comandi non vanno dati con le virgolette " " !]

```
4
3
2
1
working...
Connected to 192.168.1.210.
dd: /dev/rwd0d: Operation not supported
220 leonardo FTP server (Version 4.1 Sat Sep 7 14:31:53 CDT 2002) ready.
331 Password required for mimmo.
Password:
230-Last unsuccessful login: mar  9 set 23.10.54 2003 on /dev/pts/0 from Man
uM
230-Last login: mer 10 set 16.50.52 2003 on ftp from pascal.cicerone
230 User mimmo logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
200 Type set to I.
remote: back_nt.gz
227 Entering Passive Mode (192,168,1,210,128,88)
150 Opening data connection for back_nt.gz.
226 Transfer complete.
20 bytes sent in 00:00 (2.42 KB/s)
221 Goodbye.
#
```

Allo stesso modo per il ripristino:

- Di un intero hard disk: "slurpdisk your.ftp.server.com filename.gz"  
(Ancora una volta sottintende `rwd0d` a fine riga comando!)
- Di una singola partizione: "slurppart your.ftp.server.com filename.gz wd0e"  
(Vedi commento clonazione singola partizione)

Sostanzialmente la differenza sta nel fatto che con la singola partizione, oltre al nome dello script, cambia anche l'informazione da fornire: bisogna specificare quale partizione salvare, mentre per l'intero disco è sottinteso l'intero device, struttura di partizionamento inclusa.

Per il resto è tutto assolutamente identico.. ovviamente cambia la quantità di spazio utilizzato per l'immagine dalle due procedure!

Un'altra possibilità offerta è quella relativa al ripristino di un'immagine di un disco (come di una partizione), in una locazione differente da quella originale (o sottintesa). Ancora una volta basta specificare a fine riga comando la destinazione relativa..

Ad esempio per ripristinare l'immagine di un disco intero sul secondo disco della catena scsi, il comando `slurpdisk` avrà come ultimo parametro "`sd1`"; idem per cambiare la destinazione di una

partizione, basta cambiare l'indirizzo specificato dall'ultimo parametro di `slurppart`.

Per il resto l'indipendenza della copia dal S.O. o dal fs contenuto sul disco/parte di disco è garantito dal coltellino svizzero della replica dei dati (*dd*), che sicuramente avrà già tirato via di impiccio i vari amministratori di sistema \*nix e non solo, almeno una volta..

A proposito di *dd*, è ovviamente lui il motivo per cui le immagini dei dischi e delle partizioni sono così grandi..

Non essendo possibile accertarsi puntualmente del contenuto di un fs, *dd* lo replica per intero anche se vuoto per tre/quarti. Per evitare spreco di spazio sul server delle immagini, si potrebbero mettere a zero tutti i cilindri e le testine delle porzioni di fs non utilizzate sul “disco da salvare”, in modo che la compressione di dati “vuoti” sia la più efficace possibile. Per i sistemi unix questo può essere fatto semplicemente: basta riempire il fs di file “vuoti” e poi rimuoverli. In questo modo i dati “zero” contenuti sul disco verranno compressi al massimo.

```
dd if=/dev/zero of=/0bits bs=20971520 # bs=20m  
rm /0bits
```

Per i fs di tipo fat/ntfs è invece necessario far girare un programmino (disponibile sul sito precedentemente indicato), che fa esattamente la stessa operazione; ne sono presenti sia una versione in perl che una in pascal.. a voi la scelta! ;-)

Per finire, qualche parola sulla possibilità di ripristinare le immagini su dischi/parti di disco di dimensioni differenti dall'originale: se le dimensioni sono minori, rischiate di perdere i dati che non saranno “contenuti” sul supporto fisico, o di perdere quelli della partizione sottostante (adiacente), che verranno sovrascritti dal programma. Se le dimensioni del disco che riceverà l'immagine sono invece maggiori, potete sempre utilizzare lo spazio che avanza creando altre partizioni o per altri usi..

Ovviamente questo onere è a vs. carico, come lo sarebbe stato in ogni caso..

Ma questo.. è un altro “documento”.. ;-)

Per ora è tutto!

Arrivederci alla prossima puntata de:

COME LO FECI..