Linux - Device Mapper e oltre

- Introduzione – device “virtuali”
- Volumi logici
- Architettura del kernel
- Device Mapper e RAID a confronto
- Device Mapper
  - Rimappatura a caldo
  - dm-snapshot in dettaglio
  - Usi pratici di dm-snapshot
  - Dimostrazione dal vivo
- Network block devices
- Links

Introduzione

Sui sistemi Unix – incluso Linux – tutto o quasi è un file:
anche i dispositivi a blocchi (block devices)
cioè i dischi fissi, CD e DVD, penne USB, floppy...

Chiameremo block devices “virtuali” quelli non direttamente corrispondenti and una parte hardware del computer stesso:

– dischi RAM
– loop devices (per trasformare file in block devices)
– dischi RAID
– volumi logici (tramite device-mapper o LVM2)
– volumi criptati (dm-crypt o cryptoloop)
– userspace/network block devices
– eccetera...

Tratteremo i ***volumi logici*** e ***volumi criptati*** nel kernel Linux 2.6, più una breve panoramica su ***userspace/network block devices***

Volumi logici

Nel kernel Linux 2.6, “volume logico” significa:

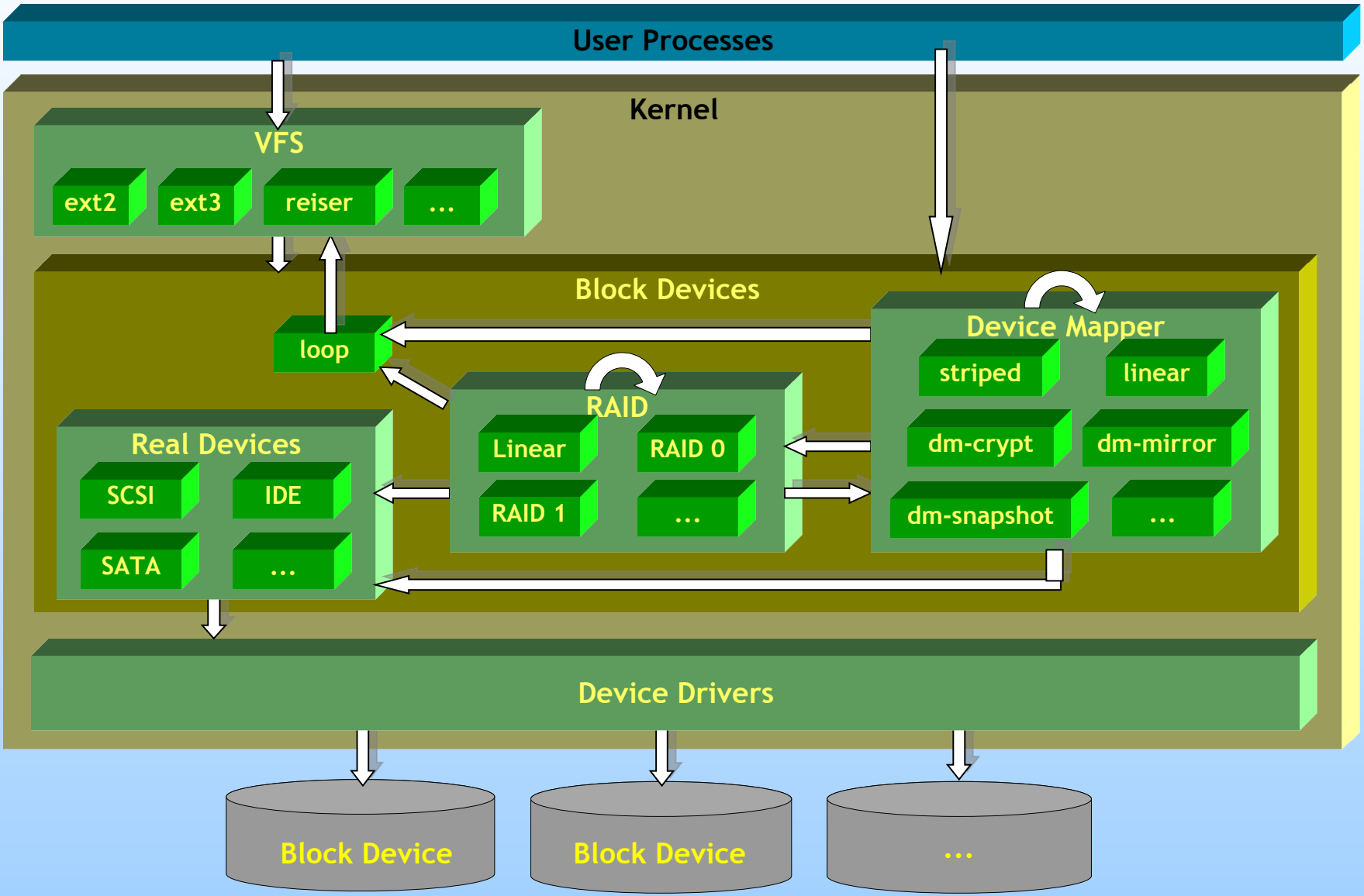
un block device virtuale, costruito a partire da una sequenza arbitraria di settori di altri block devices (reali o virtuali), utilizzando una qualche regola di composizione

Il sottosistema ***device-mapper*** si occupa di gestire i volumi logici

- può essere usato direttamente con i comandi ***dmsetup*** e ***cryptsetup***
- è la base su cui si appoggiano gestori di dischi logici più sofisticati come LVM2 e EVMS

- supporta diverse regole di composizione:
  - concatenazione (***linear***) - simile al RAID Linear
  - alternanza (***striped***) - simile al RAID 0
  - duplicazione (***dm-mirror***) - simile al RAID 1
  - volume criptato (***dm-crypt***) - sostituisce l'obsoleto ***cryptoloop***
  - istantanea (***dm-snapshot***) - crea una copia C.O.W. di un device
  - multipercorso (***dm-multipath***) - solo per hardware multipath
  - più altre regole utili in casi particolari (***dm-zero***, ***dm-delay***)

Architettura del kernel



Device Mapper e RAID a confronto

Anche se alcune regole di composizione del ***device-mapper*** sono simili a quelle dei vari livelli ***RAID*** (Linear, 0, 1...), tra i due sottosistemi ci sono differenze fondamentali:

- è possibile sospendere un volume logico e farlo ripartire in seguito
- device-mapper è nato per offrire la massima flessibilità e dinamicità: le regole di composizione possono essere modificate “a caldo” e in modo arbitrario anche mentre un volume logico è in uso
- per contro, un device virtuale RAID permette poche o nessuna modifica “a caldo”, ed è difficile da modificare strutturalmente anche quando non è in uso
- device-mapper offre poca o nessuna protezione contro i fallimenti hardware (guasti)
- al contrario, molti dei livelli RAID memorizzano i dati in modo ridondante e permettono di sostituire “a caldo” un disco rotto senza perdere dati

Il kernel Linux non fa differenza tra device reali e device virtuali: sia device-mapper che RAID possono utilizzarli entrambi.
Cioè device-mapper e RAID si possono comporre tra loro!

Rimappatura a caldo

Q: “Abbiamo capito... ma non sembra niente di eccezionale”
A: “Non abbiate fretta, il bello deve arrivare!”

L'infrastruttura presentata è molto generale e astratta,
al punto che è facile “perdersi” e lasciarsi sfuggire gli aspetti chiave:

le regole di composizione possono essere modificate a caldo

cioè si può modificare o sostituire al volo il block device usato da un filesystem!

è possibile sospendere un volume logico e farlo ripartire in seguito

dmsetup suspend blocca tutti i processi che leggono o scrivono su un volume, finché il volume non viene fatto ripartire con ***dmsetup resume***

componendo queste due funzionalità, è possibile modificare o sostituire al volo un block device in uso, in modo atomico e trasparente ai processi

dm-snapshot in dettaglio

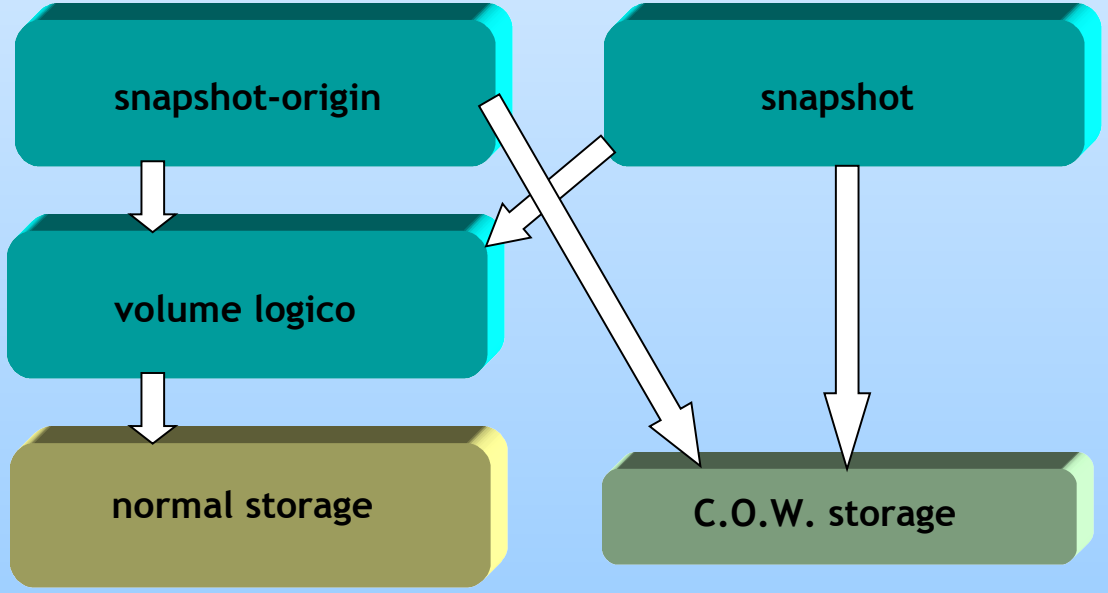
Le regole di composizione ***linear***, ***striped***, ***dm-mirror***, ***dm-zero*** e ***dm-crypt*** sono abbastanza auto-esplicative.

Invece ***dm-snapshot*** merita di essere vista più in dettaglio:

È composto da due regole, ***snapshot*** e ***snapshot-origin***

Offrono due viste modificabili indipendentemente dello stesso device

Entrambe si appoggiano ad un device aggiuntivo (C.O.W. storage) per salvare le differenze relative... ma con una differenza fondamentale!



Usi pratici di dm-snapshot

– Backup

Normalmente i backup si eseguono in single-user mode.
Eseguire il backup di un sistema “vivo” ha il rischio di ottenere files e directory incompleti o inconsistenti e timestamps errati: i files e le directory vengono modificati durante il backup!

– Sicurezza

Si può creare uno snapshot prima di installare o usare programmi di cui non ci si fida. Se il programma fa danni o non ci piace, possiamo confrontare il filesystem modificato con lo snapshot, trovare le differenze ed eventualmente buttarle e tornare allo stato iniziale

– CD-ROM

o altri dispositivi in sola lettura... possono essere affiancati da un device da usare come C.O.W. storage per farli apparire scrivibili. Volendo, le modifiche possono essere disfatte (es. al reboot) semplicemente azzerando o non usando il C.O.W. storage (limitazione: il filesystem usato deve supportare la scrittura)

Dimostrazione dal vivo

Network block device

Brevissimamente:

– NBD, analogamente a FUSE, permette di implementare in user space un servizio tipico del kernel: in questo caso un block device

– addirittura è diviso in due parti connesse tramite socket:
il client che si registra come block device
e il server a cui vengono inoltrati i comandi e che si occupa di leggere e scrivere i dati “da qualche parte”

– assieme permettono di esportare in remoto un block device

– possono essere modificati facilmente per crearsi da soli un block device semplice o complesso quanto si vuole
(block device su gmail ?)

Limitazioni:

– in passato erano segnalati possibili deadlock se client e server si trovano sulla stessa macchina.
Lo stato attuale del problema non è molto chiaro: di solito funziona lo stesso, ma in linea di principio il deadlock rimane

Links

Right To Your Own Devices - dmsetup tutorial
http://linuxgazette.net/114/kapil.html

device-mapper http://sourceware.org/dm

dm-crypt & cryptsetup http://www.saout.de/misc/dm-crypt

LVM2 Resource Page http://sourceware.org/lvm2

LVM HOWTO http://www.tldp.org/HOWTO/LVM-HOWTO/index.html

Network Block Device http://nbd.sourceforge.net/

e naturalmente i sorgenti del kernel!