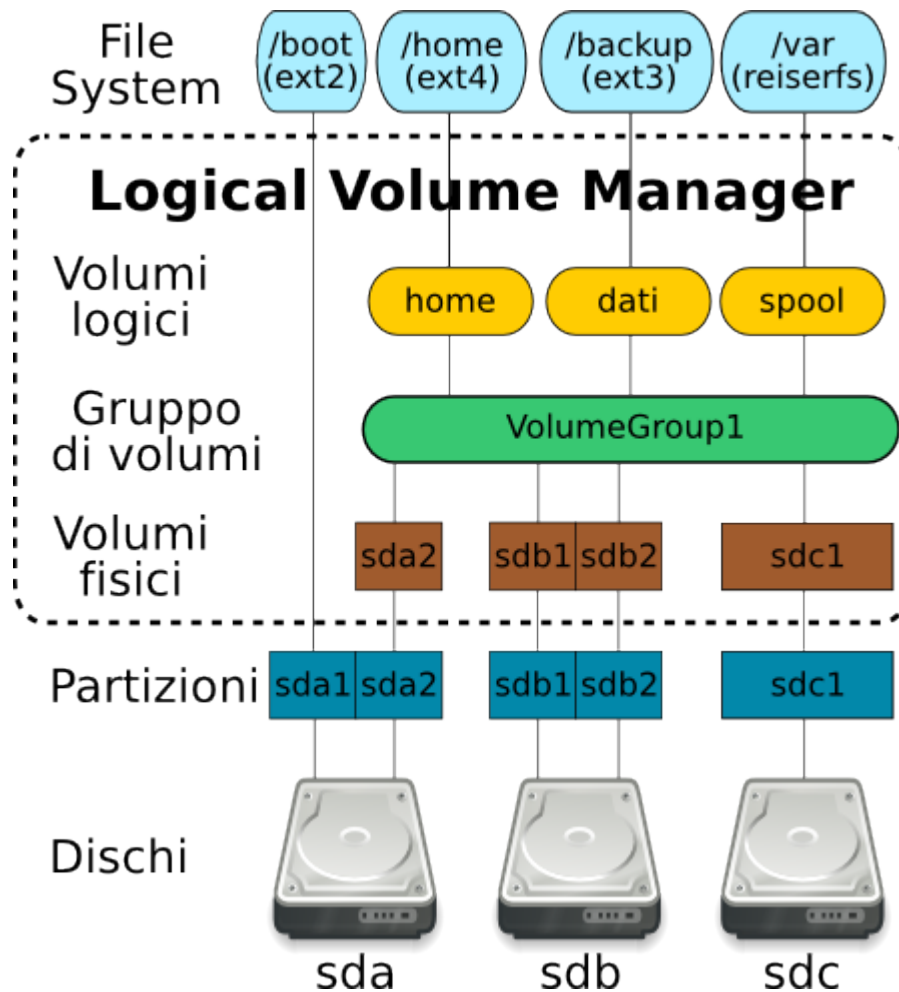


## Cos'è e come funziona LVM?



Il classico partizionamento usato nella maggior parte delle installazioni casalinghe è quello basato su *MBR (Master Boot Record)* e tabella delle partizioni MSDOS.

Con questo sistema è possibile avere un massimo di **4 partizioni primarie** oppure fino a **3 partizioni primarie**, una partizione **estesa** e all'interno di quest'ultima infinite **partizioni logiche**.

Il limite principale di questa architettura è dato dalla rigidità di questa organizzazione delle risorse. Le partizioni sono fisse e non possono essere separate in più dischi, non possono essere modificate senza smontarle e lo spazio per il file system è quello prefissato, non può né crescere né diminuire senza smontare tutto e rifare le partizioni.

Immaginando un'installazione multiutente, se il disco viene completamente riempito l'unica soluzione è quella di acquistare un disco nuovo e travasare i dati in quello, reimpostando i permessi com'erano in origine con elevati costi di tempo, oppure spostare i dati altrove, soluzione ovviamente non facilmente applicabile in multiutenza.

**LVM (Logical Volume Manager)** rivoluziona il concetto di organizzazione dello spazio fornendo un sistema di allocazione dinamica dello spazio.

## Struttura di LVM

La figura qui sopra mostra il funzionamento generale di LVM in presenza di vari dischi.

Ciascun **disco fisico** contiene al suo interno un certo numero di **partizioni**; queste possono essere usate normalmente per montare un file system (come nel caso della sda1 che punta a /boot senza far parte di LVM) oppure possono essere usate in LVM.

Ogni partizione che viene inserita in LVM prende il nome di **volume fisico** (*PV = Physical Volume*).

Tutti o solo alcuni volumi fisici possono essere assegnati a un **gruppo di volumi** (*VG = Volume group*). Questo consente di sfruttare più dischi/partizioni unite in un'unica struttura dati. L'uso di hardware dedicato migliora notevolmente il funzionamento, consentendo l'utilizzo sincrono di tutti i dischi collegati.

La somma di tutte le partizioni dedicate al gruppo di volumi definisce la dimensione del gruppo; in questo modo quando lo spazio si esaurirà basterà acquistare un nuovo disco, aggiungerlo al gruppo e la sua capienza verrà sommata allo spazio del gruppo. In questo modo si assicura la scalabilità anche su enormi dimensioni.

Tutto lo spazio dedicato a un gruppo di volumi può essere suddiviso in uno o più **volumi logici** (*LV = Logical Volume*) che verranno usati dal sistema per costituire il **file system**, semplicemente formattandoli nella maniera preferita, come se fossero delle partizioni.

## Vantaggi dell'uso di LVM

I vantaggi di utilizzare LVM rispetto un partizionamento classico sono numerosi:

1. Capacità totale espandibile senza spostamento dei dati
2. Allocazione dinamica dello spazio senza riformattazione, inclusa l'aggiunta, rimozione e sostituzione dei dischi
3. Associazione di nomi a un gruppo di dispositivi, senza preoccuparsi di quali partizioni vengano usate per fornire un gruppo di volumi
4. Ripartizione (striping) dei dati su più dischi
5. Clonazione (mirroring) dei volumi
6. Creazione di istantanee (snapshot) di un volume logico per avere delle copie dei dati uguali al momento dell'istantanea

# LVM e gestione dei volumi con Linux

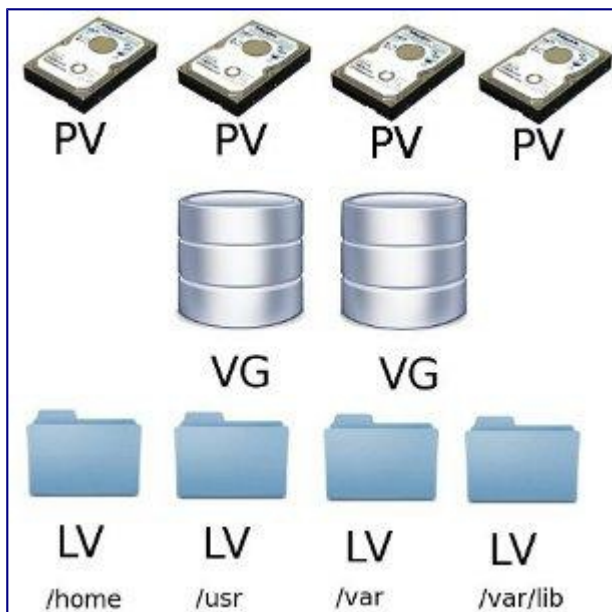
**Lavorando con Linux prima o poi ci si può trovare a dover affrontare LVM. LVM è l'acronimo di Linux Volume Manager, lo strumento messo a disposizione dei sistemi open source come piattaforma per lo storage. La caratteristica fondamentale di questo strumento è la creazione e gestione dei volumi all'interno di una macchina Linux. LVM è uno strumento affidabile e scalabile, in quanto consente di integrare diverse unità a disco, creando diversi dischi logici che possono essere, spostati e ridimensionati in modo del tutto trasparente per l'utente e senza dover bloccare le attività del sistema.**

Lavorando con Linux prima o poi ci si può trovare a dover affrontare **LVM**. LVM è l'acronimo di **Linux Volume Manager**, lo strumento messo a disposizione dei sistemi open source come piattaforma per lo storage. La caratteristica fondamentale di questo strumento **è la creazione e gestione dei volumi all'interno di una macchina Linux**. LVM è uno strumento affidabile e scalabile, in quanto consente di integrare diverse unità a disco, creando diversi dischi logici che possono essere, spostati e ridimensionati in modo del tutto trasparente per l'utente e senza dover bloccare le attività del sistema.

Grazie alle sue caratteristiche LVM si pone come **uno strumento fondamentale per la gestione dei volumi in ambienti cluster e su piattaforme distribuite**. LVM ha un funzionamento molto simile alle soluzioni RAID, con diverse unità fisiche che vengono riunite e considerati a gruppi.

## Come funziona LVM

Il disco fisso o una serie di dischi fissi viene allocata ad uno o più volumi fisici. Un volume fisico non può essere disteso su più di una unità. I volumi fisici sono combinati in gruppi di volume logico, **con l'eccezione della partizione /boot**, perché altrimenti il boot loader non potrebbe leggerlo. Sui gruppi sono creati i singoli volumi logici su cui installare un file system ed assegnare un mount point.



Schematizzando il funzionamento descritto, è possibile identificare così i seguenti step nella creazione di una soluzione con LVM:

- **Inizializzazione dischi:** LVM marca i dischi come suoi scrivendo le informazioni nelle singole porzioni di ogni disco chiamate 'private region'
- **Raggruppamento dischi:** definizione dei Volume Group ossia degli Array Virtuali di dischi.
- **Definizione volumi:** per ogni Volume Group è possibile definire i Logical Volume
- **Creazione file system:** per ogni Logical Volume si possono creare dei filesystem di tipo diverso

## Installazione di LVM

Il supporto per LVM deve essere compilato nel kernel, ed include un componente filesystem nel kernel, e delle utilities in user space. L'opzione del kernel da abilitare è:

### Device Drivers → Multi-device support (RAID and LVM)

[\*] Multiple devices driver support (RAID and LVM)

< > RAID support

< \* > Device mapper support

< > Crypt target support (NEW)

Le utilities in user space, possono essere installate tramite il gestore dei pacchetti abilitato per la distribuzione in uso.

In una gestione classica, senza LVM, i dischi fisici, sono divisi in partizioni, su ognuna delle quali viene montato un filesystem, mentre con LVM **i gruppi di volumi sono suddivisi in volumi logici su cui risiedono le partizioni.**

## Esempio d'uso di LVM

Prendiamo in considerazione una situazione con un server dedicato al backup, il cui disco è

suddiviso in tre partizioni:

- File video, 400 GB
- Backup documenti, 50 GB
- File condivisi, 10 Gb

Supponiamo che il sistema in considerazione **abbia 4 dischi da 250 Gb indicati con sda, sdb, sdc, sdd**. La prima operazione da fare è partizionarli usando fdisk, e specificando come tipo della partizione 8e, che corrisponde a partizioni LVM.

#### # fdisk /dev/sda

I comandi da impartire ad fdisk sono 4:

- p per stampare a monitor la tabella delle partizioni corrente
- n per creare la nuova partizione
- t per cambiare il tipo di partizione in Linux LVM (codice 8e)
- w per salvare le modifiche fatte.

Questa operazione va ripetuta per tutti i dischi.

### Creazione gruppi di volume e volumi logici

Il passo successivo è inizializzare i dischi con:

#### # pvcreate /dev/sda /dev/sdb /dev/sdc /dev/sdd

In questo modo tutte le partizioni dei dischi sono pronte per LVM, garantendo la creazione dei gruppi di volume. Per creare un unico gruppo di volumi, il cui nome è **volgrp**:

#### # vgcreate volgrp /dev/sda1 /dev/sdb1 /dev/sdc1 /dev/sdd1

questo comando prende come parametri le singole partizioni create sui dischi.

La creazione dei volumi logici avviene con i comandi

- # lvcreate -name medialv -size 400GB
- # lvcreate -name backuplv -size 50GB
- # lvcreate -name sharelv -size 10GB

In questo esempio, è evidente che non tutto lo spazio disponibile, viene utilizzato, in quanto una buona strategia è quella di allocare solo lo spazio che si pensa potrà servire, lasciando il rimanente non allocato, per quando sarà più chiaro che uso farne.

### Test delle operazioni

I comandi **pvdisplay** e **vgdisplay**, consentono di visualizzare i volumi fisici a disposizione di LVM ed i gruppi di volumi. In particolare vgdisplay riporta lo spazio libero mediante **PE ovvero Physical Extension**. I dischi infatti vengono suddivisi in chunk di dati (Physical Extension). Con l'opzione verbose è possibile capire quanti dischi fisici reali siano assegnati a tale volume:

#### # vgdisplay -v

Per verificare i volumi logici realizzati, il comando è **lvdisplay**. Il comando **lvdisplay** fornisce tutte le informazioni sui singoli volumi logici attivi sulla macchina:

- LV Name = Il percorso fisico e il nome del Volume Logico
- VG Name = Il nome del Volume Group ovvero il gruppo su cui il volume logico è associato
- LV UUID = Il codice univoco che identifica il volume logico
- LV Write Access = La modalità di lavoro del volume
- LV Status = Status del volume logico
- LV Size = Dimensione del Volume
- Current LE = I dati dei volumi logici possono essere mappati sui device fisici (PV) in modalità lineare o in modalità striped.

## Formattazione del Mount Point

L'ultimo passo della procedura di settaggio di LVM è **la formattazione ed il montaggio dei volumi logici creati**. Prendiamo in considerazione la necessità di utilizzare un filesystem EXT3, e di volerlo montare nella directory

**/home/client/lvm.**

Per formattare il volume **medialv** il comando è:

```
# mkfs.ext3 /dev/volgrp/medialv
```

Completato il processo di formattazione, è necessario creare la directory, che verrà usata dal sistema come punto di mount dove montare il filesystem. Il comando da eseguire è:

```
# mkdir /home/client/lvm
```

Questo comando suppone che il path fino alla directory client esista, altrimenti è possibile usare il comando **mkdir -pv /home/client/lvm** che crea ricorsivamente tutte le directories mancanti.

Alla fine occorre montare il filesystem. La procedura più semplice è quella di **aggiungere una riga nel file /etc/fstab**, in modo che ad ogni riavvio del sistema, l'operazione di mount sia automatica e non demandata ogni volta all'utente:

```
/dev/volgrp/medialv /home/client/lvm ext3 defaults 0 0
```

Aggiunta la riga al file il montaggio avviene con il comando

```
# mount /home/client/lvm
```

Per verificare l'esito dell'operazione è necessario visualizzare l'elenco dei filesystem montati e cercare l'ultimo aggiunto con il comando:

```
# df -k
```

LVM, tra le altre caratteristiche garantisce **la possibilità di estendere le unità create e funzionanti**, ridimensionando o ingrandendo in modo del tutto trasparente e senza dover resettare il sistema. In effetti prendiamo in considerazione una situazione in cui è stato aggiunto un quinto disco, sde, e che questo disco sia già stato preparato secondo la procedura già mostrata. I passi da seguire sono molto semplici:

```
# vgextend volgrp /dev/sde1
```

Volume group “volgrp ” successfully extended

```
# vgdisplay -v volgrp
```

Supponiamo che il Valore Free PE / Size riporti 100 GB disponibili per questo volume.

Aggiungiamo la metà di questo spazio, e 50 GB al volume logico medialv.

```
# lvextend -L+50G /dev/volgrp/medialv
```

È evidente come tuttavia, visualizzando lo stato del gruppo di volumi volgrp, questo riporterà uno spazio disponibile ridotto di 50 GB perché sono stati assegnati ad un volume logico

```
# vgdisplay -v volgrp
```

## Conclusioni

L'utilità di LVM a questo punto appare evidente; un esempio può essere il reimpiego di vecchi dischi, di dimensioni eterogenee o anche piccole. Messi insieme possono raggiungere una capacità di memorizzazione tale da motivarne l'uso collettivo, per esempio per la creazione di un server di backup, o un ftp interno alla propria rete. **LVM grazie alla sua caratteristica di “virtualizzare” i dischi consente una gestione molto più semplice di sistemi di questo tipo.**

Oltre a quanto già visto, LVM consente di modificare lo spazio assegnato, in modo da coprire la totalità di quello disponibile, di ridurre la dimensione dei volumi, e rimuovere un disco a causa di guasto o inserirne uno aggiuntivo, in sintesi offre una flessibilità unica nella gestione dello storage.

# LVM: funzionamento

## Guida all'LVM

### SOMMARIO

#### Parte I

##### Introduzione all'LVM

1. [Introduzione all'LVM](#)
2. **Come funziona**
3. [Gestire i VG](#)
4. [Gestire i LV](#)
5. [Gestire i PD](#)

#### Parte II

##### Casi speciali

1. [Il Boot e l'LVM](#)
2. [Installazione di Debian con l'LVM](#)
3. [Impostare un mirror LVM](#)

4. [Lavorare con gli snapshot](#)
5. [Esempi](#)

## Appendice

1. [Comandi utili](#)
2. [Sitografia](#)

In questa sezione sono introdotti i principi elementi che compongono l'*LVM* ed è introdotto il funzionamento dello stesso.

## La struttura base dell'*LVM*

Il LVM basa il proprio funzionamento sulla seguente struttura:

### *Volume Group (VG)*

trattasi del più alto livello di astrazione che raggruppa in sé una collezione di Logical Volume e di Physical Device.

### *Physical Device (PD)*

sono i nostri supporti di memorizzazione e possono essere di qualsiasi tipo, sia Hard Disk, sia nastri, sia memorie solide ma anche ambienti RAID, etc.

### *Logical Volume (LV)*

l'equivalente nel nostro *LVM* in tutto e per tutto di quella che è una partizione su un Hard Disk. L'unica piccola differenza subito evidente è l'etichetta/label (ad es. /dev/nome-VG/nome-LV invece del canonico /dev/sda1). Come ogni normale partizione il *LV* può contenere un file system a nostro piacimento.

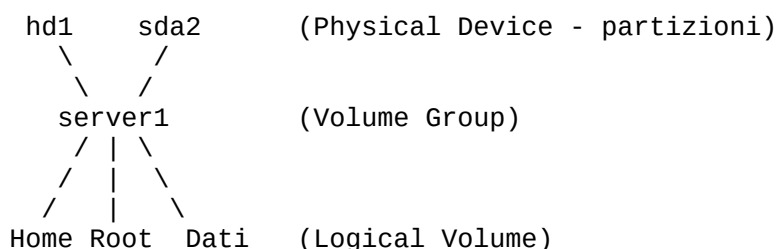
### *Logical Extend (LE)*

seppure difficilmente avremo a che farci direttamente, i *LE* sono i pezzetti in cui viene suddiviso il *LV* in modo di poterlo allocare tra i vari *PE* disponibili al *VG*.

### *Physycal Extent (PE)*

ad esempio, quando vogliamo aggiungere un *PD* al nostro *VG*, dobbiamo per prima cosa iniziarlo con il comando `pvcreeate`. Questo comando altro non fa che dividere il supporto in tanti spazi in modo da poter poi creare una corrispondenza tra i *LE* ed i *PE* e così allocare lo spazio necessario per i *LV*.

In modo un po' stilizzato, possiamo descrivere il passaggio dal concetto di partizioni a quello di *LV* nel seguente modo:



### In definitiva:

Nonostante i concetti dominanti della terminologia di LVM siano 5, come descritto precedentemente, **quelli con cui si trova più a contatto l'utente sono i primi tre: Physical Device, Volume Group e Logical Volume**. Gli altri due sono, per così dire, concetti più di basso livello, quindi è possibile approfondirli in un secondo tempo.



Come si evince dalla figura precedente, **l'elemento innovativo (che quindi conferisce maggiore flessibilità ad LVM) è il Volume Group**.

Esso in pratica fonde i volumi fisici disponibili (partizioni varie, raid, etc.) in una specie di "massa grezza" pronta per essere manipolata al fine di creare uno o più Logical Volume.

Questi ultimi, come già detto, svolgono il ruolo delle partizioni quindi sono disponibili per essere innestati nel sistema in specifici [mountpoint](#) ma in una maniera più flessibile (in termini di espandibilità e riduzione di spazio) rispetto alle partizioni tradizionali.

## Approfondimenti

La spiegazione di come si comporta e funziona l'LVM è già stata tratta nell'[e-zine numero 5](#) di [debianizzati.org](#) nello sviluppo dell'articolo [LVM mirror](#). Ecco a seguire [il riporto dell'Introduzione teorica all'LVM](#):



La prima [e-zine](#) italiana sul mondo Debian

LVM è l'acronimo di **Logical Volume Manager**, ovvero di Gestore dei Volumi Logici di Linux. Ma quindi cosa sono questi *volumi logici*?

L'idea di fondo è di sostituire al concetto di partizione un diverso contenitore (il **Logical Volume**) il quale non dipende più dalle limitate informazioni contenibili nell'[MBR](#) ma, tramite la gestione del kernel Linux, si può liberamente spostare/raggruppare/estendere da un settore all'altro di ogni disco, quasi come se fosse un file o una cartella all'interno del *file system*.

All'atto pratico [la modalità di funzionamento dell'LVM](#) può essere riassunta nel seguente modo. L'unità principale sulla quale si basa il sistema è il **Volume Group**. Detto VG verrà da noi dotato di una certa capacità di memoria pari alla capacità di tutti i supporti di massa che gli assegneremo. Tali supporti possono essere dischi SATA, PATA, USB, configurazioni RAID, SSD o quant'altro e mi riferirò ad essi come **Physical Device**. I *PD* possono anche non avere alcuna partizione, seppure si consiglia di dotarli almeno di una partizione primaria siccome altri sistemi operativi, qualora non riscontrino alcuna partizione, potrebbero inavvertitamente sovrascriverli.

I *PD*, quando sono assegnati ad un VG, sono gestiti dall'LVM suddividendo lo spazio disponibile su ognuno di essi tramite una suddivisione in unità fisiche ridotte (di default di qualche MB) dette **Physical Extent**. Per questo motivo, prima di aggiungere un *PD* ad un VG sarà necessario inizializzarlo proprio per creare su di esso i vari *PE*.

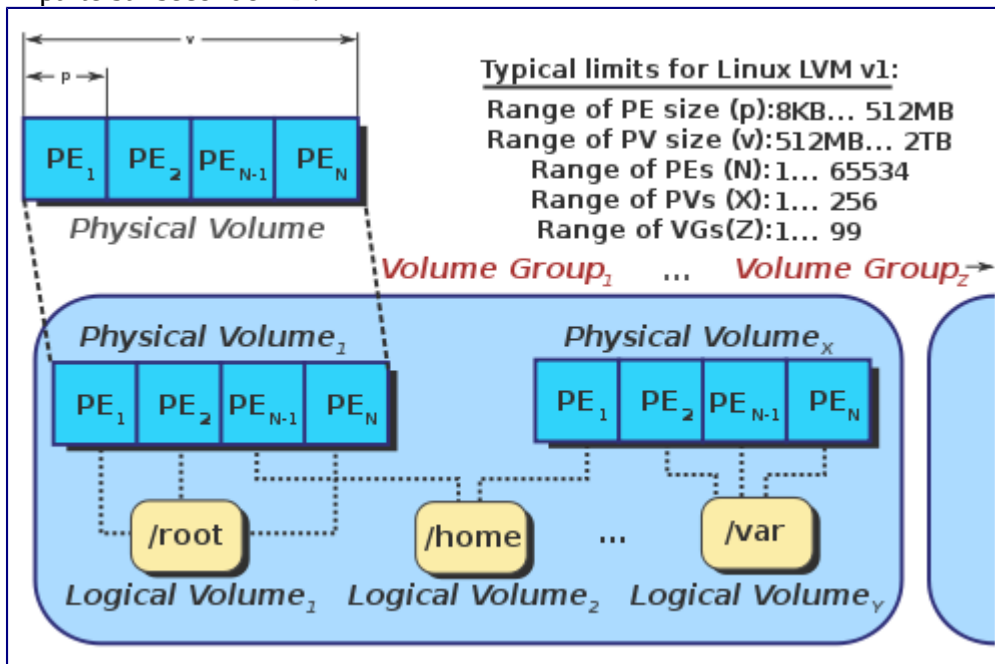
Quando creeremo un **Logical Volume** (ovvero lo spazio che useremo come "partizione") l'LVM gestirà lo spazio virtuale (logico) da noi richiesto suddividendolo in tanti **Logical Extent** tali da permettere di distribuire ogni *LE*, che altro non è che un pezzo del nostro LV, tra i *PD*. Questa allocazione avviene facendo corrispondere ad ogni *LE* un determinato *PE* secondo necessità.

Da tale modo di spezzettare i LV ed i PD nasce tutta la potenzialità dell'LVM. **Infatti, quando all'interno del nostro VG creeremo un LV l'LVM non sarà obbligato a distribuire lo spazio da noi richiesto tra i PD ma tra le PE**. Pertanto sarà possibile spostare/allargare/ridurre il LV semplicemente agendo sulla distribuzione delle *LE* tra i *PE* disponibili.

Ad esempio, quando in questo articolo creeremo il nostro LV in modalità mirrored, l'LVM si limiterà semplicemente a sincronizzare alcune *PE* di un *PD* con le *PE* del nostro secondo *PD*. Pertanto ad ogni coppia di *PE* sincronizzata corrisponderà il medesimo *LE* (e quindi il medesimo LV).

Per chi, perso tra queste sigle, volesse maggiormente approfondire segnalo alcuni utili HOWTO [\(2\)](#) [\(3\)](#).

Chi è riuscito a seguire fin qui il discorso, ecco un grafico esemplificativo del funzionamento. In questo caso i *LE* sono omessi in quanto corrispondono 1:1 ai *PE*, ovviamente se vi ci fosse un *mirror* la corrispondenza tra *LE* e *PE* sarebbe 1:2 (un *LV* fa riferimento a due diversi spazi sui *PD*). Infine si osservi il *LV* /home, quest'ultimo è in modalità *striped* ovvero è in parte sul primo *PD* ed in parte sul secondo *PD*.



## Introduzione a Logical Volume Manager (LVM)

Un Logical Volume Manager permette ad un sistema operativo di utilizzare dei dispositivi logici virtuali per l'accesso al disco che mascherano la natura dei dispositivi fisici (hard disk) su cui risiedono i dati.

In pratica è possibile utilizzare un device virtuale di cui possono fare parte diversi hard disk fisici, anche di natura, velocità e dimensioni diverse.

I vantaggi di un simile approccio alla gestione del file systems sono maggiore scalabilità, alta disponibilità del servizio, ridondanza e in alcuni casi migliori prestazioni.

L'implementazione LVM su Linux (dalla version 2.4 è presente nel thread ufficiale del kernel) è stata sviluppata dalla software house Sistina e mette a disposizione features quali:

- Possibilità di modificare i volumi logici online, senza interruzione del servizio (utile, ad esempio, per ampliare una partizione virtuale);
- Gestione di molteplici gruppi di volumi logici;
- Funzionalità di mirroring, striping o concatenating;
- Possibilità di eseguire snapshot in tempo reale senza smontare la partizione (fondamentale per operazioni di backup);

- Strumenti di gestione in command line semplici da usare;
- Supporto per software di High availability.

I vantaggi di usare un sistema LVM esistono sia su sistemi piccoli che grandi:

- su sistemi di piccole dimensioni (workstation o server con pochi hard disk) LVM permette di gestire più agilmente lo spazio su disco: ridimensionare una partizione fisica troppo piena di dati è problematico e pericoloso, mentre è semplice ed immediato cambiare le dimensioni di un volume logico.
- su sistemi di grandi dimensioni, con numerosi dischi fissi da gestire, è possibile ottimizzare meglio lo storage, modificarne le dimensioni o sostituire dischi guasti senza interrompere il servizio.

Le componenti basilari di LVM sono:

- **Volume Group (VG)** I gruppi di volume, sono il livello di astrazione più alta, contengono volumi fisici e logici.
- **Physical Volume (PV)** Di fatto sono i singoli hard disk o le loro partizioni, ogni HD o partizione è un PV (ma anche un /dev/md0 lo può essere, in quanto si presenta come device autonomo e singolo)
- **Logical Volume (LV)** Si presenta come un device a blocchi standard, è paragonabile ad una partizione di un hard disk ed è quello che viene referenziato come mount device per una data directory e che va formattato con il filesystem che si desidera.
- **Physical Extent (PE)** Ogni volume fisico viene diviso in blocchi di dati di dimensione fissa, chiamati "physical extent". Questi sono associati (1 a 1) con i:
- **Logical Extent (LE)** I blocchi di dati, della stessa dimensione, con cui è diviso un volume logico. Diversi volumi logici possono avere LE di dimensioni diverse, ma sempre uguali all'interno dello stesso volume.

In termini pratici, per mettere in piedi un sistema basato su LVM:

- Si creano dei volumi fisici, definendoli sulla base di hard disk o partizioni esistenti con il comando **pvccreate**
- Si crea un volume group, con due o più volumi fisici, con il comando **vgcreate**. Se in tempi successivi si devono aggiungere nuovi volumi fisici al gruppo si può usare il comando **vgextend**
- In un volume group esistente si possono creare i volumi logici veri e propri, con il comando **lvcreate**
- Il volume logico creato si presenta come un device a blocchi con nome tipo /dev/nome\_vg/nome\_lv, che può essere normalmente formattato con il filesystem che si desidera (ext2, ext3, reiserfs...) e montato nella directory che si desidera (/home, /var ...) come una normale partizione.

In un Volume Group possono esistere svariati Logical Volume che possono essere facilmente ridimensionati, rimossi, creati, rinominati. Facendo un paragone con un normale hard disk e la logica con cui viene usato su un sistema Unix, i LV è come se fossero delle partizioni "flessibili" di un disco fisso (il VG) a "capienza variabile", in quanto composto da un numero modificabile di dischi fisici.

La mappatura fra Physical Extent e Logical Extent, le unità di storage minime, può essere impostata in due diversi modi:

- **Linear.** I LE di un LV vengono mappati sequenzialmente ai PE di uno o più PV. In pratica un volume logico composto da due volumi fisici, avrà, ad esempio, i LE da 1 a 100 che coincidono con i PE da 1 a 100 (sempre di uguali dimensioni) sul primo volume fisico e i LE da 101 a 200 che coincidono con i PE da 1 a 100 sul secondo volume fisico.
- **Striped.** Analogamente ad un Raid 1, i LE verranno mappati alternativamente su PE di diversi volumi fisici. In questo modo le performance potrebbero migliorare (in particolare se si usano volumi fisici su bus diversi) ma una volta impostati inizialmente, con questo metodo non è più possibile aggiungere volumi fisici ad un volume logico.

Molte distribuzioni presentano LVM come un modulo del kernel: `lvmod` (Per caricarlo, nel caso non avvenga automaticamente: `modprobe lvmod`) e provvedono all'avvio, nel caso in cui sia configurato un volume group, ad eseguire gli script di setup:

`vgscan` Esegue uno scan di tutti i dischi in cerca di volume groups e crea i database con le impostazioni di base per tutti gli altri comandi lvm: `/etc/lvmtab` e `directoy /etc/lvmtab.d/`

`vgchange -ay` Attiva i volume group conosciuti dal sistema.

Allo shutdown del sistema è opportuno dare il comando (già previsto nelle distribuzioni che supportano nativamente LVM):

`vgchange -an` Disattiva i volume group conosciuti dal sistema.

Per diagnosticare cosa succede è utile la directory `/proc/lvm/` con i suoi contenuti che a loro volta forniscono dati simili a quelli che si possono ottenere con comandi come `lvdisplay`, `vgdisplay` e `pvdisplay`.

di una tecnologia quasi omonima (*Logical Volume Management*) che permette di rendere più potente e flessibile la gestione dello storage. La sua funzione principale è quella di interporre uno strato software tra le unità fisiche, ovvero gli hard disk, e la ripartizione logica dello spazio di memorizzazione, cioè le unità accessibili dal sistema operativo. Grazie a questo livello intermedio si possono disaccoppiare le due entità e ottenere una flessibilità di configurazione molto superiore: per esempio, si possono assegnare più dischi a una stessa unità logica, impostare la ridondanza o lo striping, aumentare lo spazio dedicato a ogni partizione secondo necessità e molto altro ancora. Quasi tutte le ultime versioni delle principali distribuzioni Linux supportano Lvm, ma solo in pochissimi casi lo utilizzano per default. Per gestire tutto lo storage tramite Lvm, comprese le unità di sistema, la soluzione più semplice è una nuova installazione del sistema operativo; in ogni caso, sarà necessario disporre di almeno una piccola partizione di tipo tradizionale, poiché il Bios dovrà potervi accedere per

avviare il boot. Se invece si vuole passare a Lvm soltanto per le unità dedicate allo storage, è possibile attivarlo in qualsiasi momento: vediamo come, usando Ubuntu 12.10.

Aprirete Ubuntu Software Center e scrivete *Lvm* nel campo di ricerca, a destra. Selezionate poi l'elemento *Logical Volume Management* nell'elenco dei risultati e fate clic sul pulsante *Installa* per avviare il download e l'installazione. Una volta conclusa questa procedura, chiudete Software Center e richiamate *Gestione Volumi Logici*, tramite il Dash.

L'applicazione di gestione è piuttosto intuitiva, ma è opportuno muoversi sempre con estrema cautela poiché si sta lavorando con i dati memorizzati sull'hard disk. L'elenco *Entità non inicializzate* mostra i dischi e le partizioni riconosciuti dal sistema, compresi quelli attivi: di solito il disco di sistema di Ubuntu è quello indicato con */dev/sda/*, e contiene tre partizioni di cui una dedicata allo swap. Selezionate un disco o una partizione vuota nell'elenco di sinistra, e inicializzatelo facendo clic sul pulsante *Inizializza Entità*, in basso. Confermate la volontà di inicializzare il disco, con un clic su *Sì* nella finestra di dialogo successiva. Il programma propone di creare una singola partizione per indirizzare tutto lo spazio presente sul disco: nella maggior parte dei casi si può tranquillamente accettare il consiglio.

Per creare un'unità Lvm, fate clic su *Crea nuovo Gruppo di Volumi*: nella finestra successiva basta inserire un nome per il gruppo e controllare le altre impostazioni, che possono essere lasciate ai valori di default. Selezionate la *Visualizzazione logica* del gruppo appena creato, e fate clic sul pulsante *Crea nuovo Volume Logico*. Si aprirà una finestra di dialogo che permetterà di specificare il nome del volume, la sua tipologia (Lineare, Mirrored o Striped) e la sua dimensione. Potrete anche specificare il file system da utilizzare e il punto di montaggio. Se lo spazio dovesse esaurirsi, basterà installare un nuovo hard disk, inicializzarlo, aggiungerlo al gruppo volumi e aumentare la dimensione del volume logico.