



Red Hat Enterprise Linux 6

Logical Volume Manager Administration

Guida per l'amministratore LVM

Edizione 1

Last Updated: 2017-10-12

Red Hat Enterprise Linux 6 Logical Volume Manager Administration

Guida per l'amministratore LVM

Edizione 1

Landmann

rlandmann@redhat.com

Nota Legale

Copyright © 2011 Red Hat, Inc. and others.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Sommario

Questo libro descrive il logical volume manager 'LVM', ed include le informazioni su come eseguire LVM in un ambiente clusterizzato.

Indice

INTRODUZIONE	5
1. INFORMAZIONI SU QUESTA GUIDA	5
2. A CHI È RIVOLTO	5
3. VERSIONI SOFTWARE	5
4. DOCUMENTAZIONE RELATIVA	5
5. ABBIAMO BISOGNO DI COMMENTI!	6
CAPITOLO 1. LVM LOGICAL VOLUME MANAGER	7
1.1. FUNZIONI NUOVE E MODIFICATE	7
1.1.1. Funzioni nuove e modificate di Red Hat Enterprise Linux 6.0	7
1.1.2. Funzioni nuove e modificate di Red Hat Enterprise Linux 6.1	8
1.2. VOLUMI LOGICI	8
1.3. PANORAMICA SULL'ARCHITETTURA LVM	9
1.4. LVM LOGICAL VOLUME MANAGER (CLVM)	10
1.5. PANORAMICA DEL DOCUMENTO	12
CAPITOLO 2. COMPONENTI DI LVM	14
2.1. VOLUMI FISICI	14
2.1.1. Disposizione del volume fisico LVM	14
2.1.2. Partizioni multiple su di un disco	15
2.2. GRUPPI DI VOLUMI	15
2.3. VOLUMI LOGICI LVM	16
2.3.1. Volumi lineari	16
2.3.2. Volumi logici segmentati	18
2.3.3. Volumi logici speculari	19
2.3.4. Volumi delle snapshot	20
CAPITOLO 3. PANORAMICA SULL'AMMINISTRAZIONE DI LVM	22
3.1. CREAZIONE DEI VOLUMI LVM IN UN CLUSTER	22
3.2. PANORAMICA SULLA CREAZIONE DEL VOLUME LOGICO	23
3.3. SVILUPPO DI UN FILE SYSTEM SU DI UN VOLUME LOGICO	23
3.4. BACKUP DEL VOLUME LOGICO	23
3.5. REGISTRAZIONE	24
CAPITOLO 4. AMMINISTRAZIONE DI LVM CON I COMANDI CLI	25
4.1. COME UTILIZZARE I COMANDI CLI	25
4.2. AMMINISTRAZIONE DEL VOLUME FISICO	26
4.2.1. Creazione dei volumi fisici	26
4.2.1.1. Impostazione del tipo di partizione	26
4.2.1.2. Inizializzazione dei volumi fisici	27
4.2.1.3. Scansione per dispositivi a blocchi	27
4.2.2. Visualizzazione dei volumi fisici	28
4.2.3. Come impedire l'assegnazione su di un volume fisico	29
4.2.4. Come variare la dimensione di un volume fisico	29
4.2.5. Rimozione dei volumi fisici	29
4.3. AMMINISTRAZIONE DEL GRUPPO DI VOLUMI	29
4.3.1. Creazione dei gruppi di volumi	29
4.3.2. Creazione dei gruppi di volumi in un cluster	30
4.3.3. Aggiunta di un volume fisico ad un gruppo di volumi	31
4.3.4. Come visualizzare i gruppi di volumi	31
4.3.5. Scansione dischi per i gruppi di volumi per la creazione del file di cache	32
4.3.6. Rimozione dei volumi fisici da un gruppo di volumi	32

4.3.7. Modifica dei parametri di un gruppo di volumi	33
4.3.8. Attivazione e disattivazione dei gruppi di volumi	33
4.3.9. Rimozione dei gruppi di volumi	34
4.3.10. Separazione di un gruppo di volumi	34
4.3.11. Come unire i gruppi di volumi	34
4.3.12. Esecuzione del back up dei metadati del gruppo di volumi	34
4.3.13. Come rinominare un gruppo di volumi	35
4.3.14. Come spostare un gruppo di volumi su di un altro sistema	35
4.3.15. Come ricreare una directory del gruppo di volumi	36
4.4. AMMINISTRAZIONE DEL VOLUME LOGICO	36
4.4.1. Creazione di volumi logici lineari	36
4.4.2. Creazione dei volumi segmentati	37
4.4.3. Creazione volumi speculari	38
4.4.3.1. Politica sugli errori del volume logico speculare	41
4.4.3.2. Separazione di una immagine ridondante del volume logico speculare	41
4.4.3.3. Correzione di un dispositivo logico speculare	42
4.4.3.4. Come modificare la configurazione del volume speculare	42
4.4.4. Creazione dei volumi della snapshot	43
4.4.5. Unione dei volumi della snapshot	44
4.4.6. Numeri del dispositivo persistenti	44
4.4.7. Modifica della dimensione dei volumi logici	45
4.4.8. Modifica dei parametri di un gruppo di volumi logici	45
4.4.9. Modifica del nome dei volumi logici	45
4.4.10. Rimozione dei volumi logici	45
4.4.11. Visualizzazione dei volumi logici	46
4.4.12. Come aumentare la dimensione dei volumi logici	46
4.4.12.1. Come estendere un volume segmentato	47
4.4.12.2. Estensione di un volume logico con politica di assegnazione cling	48
4.4.13. Come ridurre la dimensione dei volumi logici	50
4.5. CONTROLLO DELLE SCANSIONI DEL DISPOSITIVO LVM CON I FILTRI	50
4.6. RIPOSIZIONAMENTO DATI ONLINE	51
4.7. ATTIVAZIONE DEI VOLUMI LOGICI SU NODI INDIVIDUALI IN UN CLUSTER	52
4.8. PERSONALIZZAZIONE DEI RIPORTI PER LVM	52
4.8.1. Controllo del formato	52
4.8.2. Selezione dell'oggetto	54
Il comando pvs	55
Il comando vgs	57
Il comando lvs	58
4.8.3. Come ordinare i riporti di LVM	61
4.8.4. Come specificare le unità	62
CAPITOLO 5. ESEMPI DI CONFIGURAZIONE LVM	64
5.1. CREAZIONE DI UN VOLUME LOGICO LVM SU TRE DISCHI	64
5.1.1. Creazione dei volumi fisici	64
5.1.2. Creazione del gruppo di volumi	64
5.1.3. Creazione del volume logico	64
5.1.4. Creazione del file system	65
5.2. CREAZIONE DI UN VOLUME LOGICO SEGMENTATO	65
5.2.1. Creazione dei volumi fisici	65
5.2.2. Creazione del gruppo di volumi	66
5.2.3. Creazione del volume logico	66
5.2.4. Creazione del file system	66
5.3. SEPARAZIONE DI UN GRUPPO DI VOLUMI	67

5.3.1. Come determinare lo spazio disponibile	67
5.3.2. Come spostare i dati	67
5.3.3. Come dividere il gruppo di volumi	67
5.3.4. Creazione di un nuovo volume logico	68
5.3.5. Creazione di un file system e montaggio di un nuovo volume logico	68
5.3.6. Attivazione e montaggio del volume logico originale	68
5.4. RIMOZIONE DI UN DISCO DA UN VOLUME LOGICO	69
5.4.1. Come spostare le estensioni su volumi fisici esistenti	69
5.4.2. Come spostare le estensioni su di un nuovo disco	70
5.4.2.1. Creazione di un nuovo Physical Volume	70
5.4.2.2. Aggiungete il nuovo Physical Volume al gruppo di volumi	70
5.4.2.3. Come spostare i dati	70
5.4.2.4. Rimozione di un volume fisico vecchio dal gruppo di volumi	71
5.5. CREAZIONE DI UN VOLUME LOGICO LVM SPECULARE IN UN CLUSTER	71
CAPITOLO 6. TROUBLESHOOTING DI LVM	75
6.1. INFORMAZIONI DIAGNOSTICHE PER IL TROUBLESHOOTING	75
6.2. COME VISUALIZZARE LE INFORMAZIONI SU DISPOSITIVI FALLITI	75
6.3. PROCESSO DI RECUPERO DA UN LVM MIRROR FAILURE	76
6.4. RECUPERO DEI METADATI DEL VOLUME FISICO	79
6.5. SOSTITUZIONE DI UN VOLUME FISICO MANCANTE	81
6.6. RIMOZIONE DEI VOLUMI FISICI PERSI DA UN GRUPPO DI VOLUMI	81
6.7. ESTENSIONI DISPONIBILI INSUFFICIENTI PER UN VOLUME LOGICO	82
CAPITOLO 7. AMMINISTRAZIONE DI LVM CON LA GUI DI LVM	83
APPENDICE A. DEVICE MAPPER	84
A.1. TABELLA DI MAPPATURA DEL DISPOSITIVO	84
A.1.1. Target di mappatura lineare	85
A.1.2. Target per la mappatura segmentata	85
A.1.3. Il target di mappatura mirror	87
A.1.4. Target di mappatura snapshot e snapshot-origin	89
A.1.5. Target di mappatura 'error'	91
A.1.6. Target di mappatura zero	91
A.1.7. Il target di mappatura multipath	92
A.1.8. Target di mappatura crypt	94
A.2. IL COMANDO DMSETUP	95
A.2.1. Il comando dmsetup info	95
A.2.2. Il comando dmsetup ls	97
A.2.3. Il comando dmsetup status	98
A.2.4. Il comando dmsetup deps	98
A.3. SUPPORTO DEVICE MAPPER PER IL DEVICE MANAGER UDEV	99
A.3.1. Integrazione udev con il Device Mapper	99
A.3.2. Comandi ed interfacce che supportano udev	101
APPENDICE B. FILE DI CONFIGURAZIONE LVM	103
B.1. FILE DI CONFIGURAZIONE LVM	103
B.2. ESEMPIO DI FILE LVM.CONF	103
APPENDICE C. TAG OGGETTO LVM	116
C.1. COME AGGIUNGERE E RIMUOVERE I TAG DAGLI OGGETTI	116
C.2. TAG DELL'HOST	116
C.3. CONTROLLO ATTIVAZIONE CON I TAG	117
APPENDICE D. METADATI DEL GRUPPO DI VOLUMI LVM	118

D.1. ETICHETTA DEL VOLUME FISICO	118
D.2. CONTENUTI DEI METADATI	118
D.3. ESEMPIO DI METADATI	119
APPENDICE E. CRONOLOGIA DI REVISIONE	122
INDICE ANALITICO	123

INTRODUZIONE

1. INFORMAZIONI SU QUESTA GUIDA

Questa guida descrive il Logical Volume Manager (LVM), e contiene le informazioni su come eseguire LVM in un ambiente clusterizzato.

2. A CHI È RIVOLTO

Questo libro è rivolto agli amministratori che gestiscono i sistemi sui quali viene eseguito un sistema operativo Linux. Necessita di una conoscenza di Red Hat Enterprise Linux 6 e di come amministrare un file system GFS2.

3. VERSIONI SOFTWARE

Tabella 1. Versioni software

Software	Descrizione
RHEL 6	si riferisce a RHEL 6 e versioni più recenti
GFS2	si riferisce a GFS2 per RHEL 6 e versioni più recenti

4. DOCUMENTAZIONE RELATIVA

Per maggiori informazioni su come utilizzare Red Hat Enterprise Linux, si prega di consultare le seguenti risorse:

- *Installation Guide*— Fornisce le informazioni relative al processo d'installazione di Red Hat Enterprise Linux 6.
- *Deployment Guide*— Fornisce le informazioni relative all'implementazione, configurazione e amministrazione di Red Hat Enterprise Linux 6.
- *Storage Administration Guide*— Fornisce le informazioni necessarie per una gestione efficace dei dispositivi di storage e dei file system su Red Hat Enterprise Linux 6.

Per maggiori informazioni su come utilizzare High Availability Add-On e Resilient Storage Add-On per Red Hat Enterprise Linux 6, consultare le seguenti risorse:

- *Panoramica sull'High Availability Add-On*— Fornisce una panoramica molto dettagliata sul Red Hat High Availability Add-On.
- *Amministrazione del Cluster*— Fornisce le informazioni relative all'installazione, configurazione e gestione per il Red Hat High Availability Add-On.

- *Global File System 2: Configurazione e amministrazione*— Fornisce le informazioni sull'installazione, configurazione ed amministrazione del Red Hat GFS2 (Red Hat Global File System 2) incluso con il Resilient Storage Add-On.
- *DM Multipath* — Fornisce le informazioni relative all'utilizzo del Device-Mapper Multipath di Red Hat Enterprise Linux 6.
- *Amministrazione Load Balancer*— Fornisce le informazioni necessarie per la configurazione di sistemi ad elevate prestazioni con il Load Balancer Add-On, un set di componenti software integrati il quale fornisce i Linux Virtual Servers (LVS) per il bilanciamento del carico IP su di un set di server reali.
- *Note di rilascio*— Fornisce le informazioni sulla release corrente dei prodotti di Red Hat.

La documentazione relativa all'High Availability Add-On ed altre documentazioni di Red Hat sono disponibili in versione HTML, PDF, e RPM sul CD di documentazione di Red Hat Enterprise Linux e online su <http://www.redhat.com/docs/>.

5. ABBIAMO BISOGNO DI COMMENTI!

Se individuate degli errori di battitura o se pensate di poter contribuire al miglioramento di questo manuale contattateci subito. Inviare i vostri suggerimenti in Bugzilla: <http://bugzilla.redhat.com/> sul componente **Red Hat Enterprise Linux 6** e **doc-Logical_Volume_Manager**. Quando inviate un bug report assicuratevi di indicare l'identificatore del manuale:

Logical_Volume_Manager_Administration(EN)-6 (2011-05-19-15:20).

Se inviate un suggerimento per contribuire al miglioramento della documentazione cercate di essere il più specifici possibile. Se avete individuato un errore, indicate il numero della sezione e alcune righe di testo in modo da agevolare la ricerca dell'errore.

CAPITOLO 1. LVM LOGICAL VOLUME MANAGER

Questo capitolo fornisce un sommario delle nuove funzioni di LVM logical volume manager per la release iniziale di Red Hat Enterprise Linux 6, ed una panoramica dettagliata dei componenti relativi al Logical Volume Manager (LVM).

1.1. FUNZIONI NUOVE E MODIFICATE

Questa sezione elenca le funzioni nuove e quelle modificate di LVM logical volume manager incluse con la release iniziale e seguente di Red Hat Enterprise Linux 6.

1.1.1. Funzioni nuove e modificate di Red Hat Enterprise Linux 6.0

Red Hat Enterprise Linux 6.0 include le seguenti modifiche ed aggiornamenti relativi alle funzioni e documentazione.

- È possibile definire il comportamento di un volume logico speculare in presenza di un errore del dispositivo tramite i parametri `mirror_image_fault_policy` e `mirror_log_fault_policy` nella sezione `activation` del file `lvm.conf`. Dopo aver impostato i parametri su `remove`, il sistema cercherà di rimuovere il dispositivo che presenta l'errore e continuerà l'esecuzione senza di esso. Se il parametro è impostato su `allocate`, il sistema cercherà di rimuovere il dispositivo fallito ed assegnerà lo spazio su di un nuovo dispositivo come sostituto del dispositivo in questione; questa politica si comporterà come la politica `remove` se nessun dispositivo e spazio idonei possono essere assegnati come sostituti. Per informazioni sulle politiche relative all'errore del mirror di LVM consultare [Sezione 4.4.3.1, «Politica sugli errori del volume logico speculare»](#).
- Per la versione di Red Hat Enterprise Linux 6 lo stack di Linux è stato migliorato in modo da poter processare le informazioni sui limiti relativi all'I/O forniti dal rivenditore. Ciò permette ai tool di gestione per lo storage, incluso LVM, di ottimizzare l'accesso ed il posizionamento dei dati. Questo supporto può essere disabilitato modificando i valori predefiniti di `data_alignment_detection` e `data_alignment_offset_detection` nel file `lvm.conf`. A tale scopo non è consigliato disabilitare questo supporto.

Per informazioni sull'allineamento dei dati in LVM e per la modifica dei valori predefiniti di `data_alignment_detection` e `data_alignment_offset_detection`, consultare la documentazione relativa al file `/etc/lvm/lvm.conf`, presente anche in [Appendice B, File di configurazione LVM](#). Per informazioni generali sul supporto per lo stack I/O ed i relativi limiti in Red Hat Enterprise Linux 6, consultare la *Storage Administration Guide*.

- Con Red Hat Enterprise Linux 6, il Device Mapper fornisce un supporto diretto per l'integrazione di `udev`. Ciò fornisce una sincronizzazione del Device Mapper con tutta la processazione `udev` relativa ai dispositivi del Device Mapper, incluso i dispositivi LVM. Per informazioni sul supporto del Device Mapper per il device manager `udev`, consultare [Sezione A.3, «Supporto Device Mapper per il Device Manager udev»](#).
- Per la release Red Hat Enterprise Linux 6 usare il comando `lvconvert --repair` per riparare un mirror dopo un errore del disco. Tale comando ripristinerà lo stato corretto del mirror. Per informazioni sul comando `lvconvert --repair` consultare [Sezione 4.4.3.3, «Correzione di un dispositivo logico speculare»](#).
- Con la release di Red Hat Enterprise Linux 6 è possibile usare l'opzione `--merge` del comando `lvconvert` per eseguire il merge di una snapshot nel proprio volume d'origine. Per informazioni sul merge delle snapshot consultare [Sezione 4.4.5, «Unione dei volumi della](#)

[snapshot](#)».

- Con la release Red Hat Enterprise Linux 6 sarà possibile usare l'opzione `--splitmirrors` del comando `lvconvert` per separare una immagine ridondante di un volume logico speculare in modo da formare un nuovo volume logico. Per informazioni su come utilizzare questa opzione consultare [Sezione 4.4.3.2, «Separazione di una immagine ridondante del volume logico speculare»](#).
- Sarà ora possibile creare un log del mirror per un dispositivo logico speculare, il quale a sua volta sarà speculare, tramite l'opzione `--mirrorlog mirrored` del comando `lvcreate` durante la creazione di un dispositivo logico speculare. Per informazioni su come usare questa opzione consultare [Sezione 4.4.3, «Creazione volumi speculari»](#).

1.1.2. Funzioni nuove e modificate di Red Hat Enterprise Linux 6.1

Red Hat Enterprise Linux 6.1 include le seguenti modifiche ed aggiornamenti relativi alle funzioni e documentazione.

- La release Red Hat Enterprise Linux 6.1 supporta la creazione dei volumi logici delle snapshot dei volumi logici speculari. La creazione di una snapshot di un volume speculare è simile alla creazione di una snapshot di un volume logico segmentato o lineare. Per informazioni sulla creazione delle snapshot dei volumi consultare [Sezione 4.4.4, «Creazione dei volumi della snapshot»](#).
- Durante l'estensione di un volume LVM sarà possibile usare l'opzione `--alloc cling` del comando `lvextend` per specificare la politica di assegnazione `cling`. Questa politica selezionerà lo spazio sugli stessi volumi fisici come segmento ultimo del volume logico esistente. Se lo spazio sui volumi fisici non è sufficiente e l'elenco dei tag è definito nel file `lvm.conf`, LVM controllerà la presenza di tag collegati ai volumi fisici e cercherà di corrispondere i tag dei volumi fisici tra estensioni esistenti e quelle nuove.

Per informazioni su come estendere i volumi speculari LVM con l'opzione `--alloc cling` del comando `lvextend` consultare [Sezione 4.4.12.2, «Estensione di un volume logico con politica di assegnazione cling»](#).

- È ora possibile specificare argomenti multipli `--addtag` e `--deltag` all'interno di un singolo comando `pvchange`, `vgchange`, o `lvchange`. Per informazioni su come aggiungere o rimuovere i tag consultare [Sezione C.1, «Come aggiungere e rimuovere i tag dagli oggetti»](#).
- È stato esteso l'elenco di caratteri permessi in LVM dei tag dell'oggetto ed ora sarà possibile utilizzare `/`, `=`, `!`, `:`, `#`, and `&`. Per informazioni sui tag di LVM consultare [Appendice C, Tag oggetto LVM](#).
- È possibile combinare RAID0 (striping) e RAID1 (mirroring) in un unico volume logico. La creazione di un volume logico insieme alla determinazione del numero dei mirror (`--mirrors X`) ed al numero di segmenti (`--stripes Y`) genererà un dispositivo mirror con dispositivi segmentati. Per maggiori informazioni su come creare volumi logici speculari consultate [Sezione 4.4.3, «Creazione volumi speculari»](#).
- Con Red Hat Enterprise Linux 6.1 se desiderate creare un backup di dati consistente su di un volume logico clusterizzato sarà possibile attivare il volume e successivamente creare la snapshot. Per informazioni su come attivare i volumi logici su un nodo consultate [Sezione 4.7, «Attivazione dei volumi logici su nodi individuali in un cluster»](#).

1.2. VOLUMI LOGICI

La gestione del volume crea un livello di astrazione rispetto allo storage fisico il quale permette all'utente di creare i volumi di storage logici. Tale operazione fornisce maggiore flessibilità rispetto all'utilizzo dello storage fisico. Con un volume logico non sarete limitati dalle dimensioni del disco fisico. In aggiunta, la configurazione dello storage dell'hardware viene nascosta al software in modo da poter essere modificata e spostata senza alcuna interruzione delle applicazioni o senza smontare i file system. Tale processo riduce i costi operativi.

I volumi logici forniscono i seguenti vantaggi rispetto all'utilizzo diretto dello storage fisico:

- Capacità più flessibile

Quando si utilizzano i volumi logici i file system possono essere estesi attraverso dischi multipli, poichè è possibile aggregare sia i dischi che le partizioni in un volume logico singolo.

- Pool di storage ridimensionabili

È possibile estendere o ridurre la dimensione dei volumi logici con alcuni comandi software semplici, senza riformattare e ripartizionare i dispositivi del disco interessati.

- Riassegnazione dati online

Per implementare sottosistemi di storage nuovi, più veloci o più resistenti, è possibile spostare i dati mentre il vostro sistema è attivo. I dati possono essere riassegnati sui dischi durante l'utilizzo dei dischi stessi. Per esempio, è possibile svuotare un disco di tipo hot-swappable prima di rimuoverlo.

- Come nominare un dispositivo in modo conveniente

I volumi dello storage locale possono essere gestiti in gruppi definiti dall'utente, i quali possono essere chiamati a vostra discrezione.

- Disk striping

È possibile creare un volume logico in grado di scrivere i dati su due o più dischi. Ciò potrebbe aumentarne drammaticamente le prestazioni.

- Volumi in mirroring

I volumi logici forniscono un modo molto conveniente per configurare un mirror per i vostri dati.

- Snapshot del volume

Utilizzando i volumi logici è possibile eseguire delle snapshot del dispositivo per un backup conforme, oppure per provare gli effetti delle modifiche senza interessare i dati reali.

L'implementazione di queste caratteristiche in LVM viene descritta nel remainder di questo documento.

1.3. PANORAMICA SULL'ARCHITETTURA LVM

Per la release di Red Hat Enterprise Linux 4 del sistema operativo di Linux, il logical volume manager LVM1 originale è stato sostituito da LVM2, il quale presenta una struttura del kernel più generica rispetto a LVM1. LVM2 presenta i seguenti miglioramenti rispetto a LVM1:

- capacità più flessibile

- conservazione dei metadati più efficiente
- un formato migliore per il processo di ripristino
- nuovo formato per i metadati ASCII
- modifiche atomiche per i metadati
- copie ridondanti di metadati

LVM2 è compatibile con LVM1, con la sola eccezione del supporto del cluster e delle snapshot. È possibile convertire un gruppo dei volumi da un formato LVM1 ad un formato LVM2 con il comando `vgconvert`. Per informazioni su come convertire il formato dei metadati, consultate la pagina man di `vgconvert(8)`.

L'unità dello storage fisico di un volume logico LVM è un dispositivo a blocchi, come ad esempio una partizione o un disco intero. Questo dispositivo viene inizializzato come un *physical volume* (PV) di LVM.

Per creare un volume logico LVM, i volumi fisici vengono raggruppati in un *volume group* (VG). Ciò creerà uno spazio del disco dal quale è possibile assegnare i volumi logici LVM (LV). Questo processo è analogo al processo attraverso il quale i dischi vengono divisi in partizioni. Il volume logico viene utilizzato dai file system e dalle applicazioni (come ad esempio i database).

Figura 1.1, «Componenti del volume logico LVM» mostra i componenti di un volume logico semplice LVM:

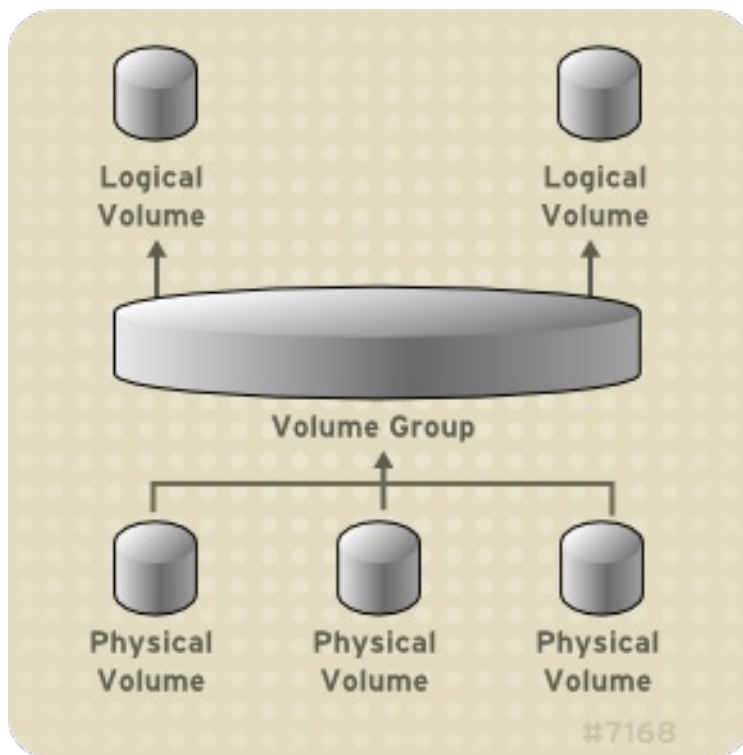


Figura 1.1. Componenti del volume logico LVM

Per informazioni dettagliate sui componenti di un volume logico LVM consultate [Capitolo 2, Componenti di LVM](#).

1.4. LVM LOGICAL VOLUME MANAGER (CLVM)

Il Clustered Logical Volume Manager (CLVM) è un set di estensioni clustering per LVM. Le suddette estensioni permettono ad un cluster di computer di gestire lo storage condiviso (per esempio, su di un SAN) utilizzando LVM. CLVM è parte della Resilient Storage Add-On.

L'utilizzo di CLVM dipende dai requisiti del vostro sistema:

- Se solo uno nodo del vostro sistema ha bisogno di accedere allo storage da voi configurato come volumi logici, allora sarà possibile utilizzare LVM senza le estensioni CLVM, in questo modo i volumi logici creati con il nodo in questione risulteranno locali al nodo.
- Se utilizzate un sistema clusterizzato per il processo di failover dove un singolo nodo accede allo storage, e risulta attivo in ogni dato momento, allora è consigliato utilizzare gli agent High Availability Logical Volume Management (HA-LVM). Per informazioni su HA-LVM, consultare la guida *Configurazione e gestione di un Red Hat Cluster*
- Se più nodi del cluster hanno la necessità di accedere al vostro storage, il quale verrà di conseguenza condiviso tra i nodi attivi, allora sarà necessario utilizzare CLVM. CLVM permette ad un utente di configurare i volumi logici su di uno storage condiviso, bloccando l'accesso allo storage fisico durante la configurazione di un volume logico ed utilizza i servizi di bloccaggio clusterizzati per gestire lo storage condiviso.

Per poter utilizzare CLVM i software High Availability Add-On e Resilient Storage Add-On, incluso il demone `clmvd`, devono essere in esecuzione. Il demone `clmvd` rappresenta l'estensione più importante del clustering per LVM. Il demone `clmvd` viene eseguito su ogni computer del cluster e distribuisce gli aggiornamenti dei metadati LVM in un cluster, presentando ad ogni computer la stessa visuale dei volumi logici. Per informazioni su come installare e amministrare High Availability Add-On consultare la *Configurazione e gestione di un Red Hat Cluster*

Per assicurarsi che `clmvd` sia stato iniziato al momento dell'avvio eseguire un comando `chkconfig ... on` sul servizio `clvmd` nel modo seguente:

```
# chkconfig clvmd on
```

Se il demone `clvmd` non è stato iniziato sarà possibile eseguire un comando `service ... start` sul servizio `clvmd` nel modo seguente:

```
# service clvmd start
```

La creazione dei volumi logici LVM in un ambiente cluster è identica alla creazione dei volumi logici LVM su di un nodo. Non vi è alcuna differenza nei comandi LVM o nell'interfaccia utente grafica LVM come riportato in [Capitolo 4, Amministrazione di LVM con i comandi CL](#) e [Capitolo 7, Amministrazione di LVM con la GUI di LVM](#). Per poter abilitare i volumi LVM da voi creati in un cluster, l'infrastruttura stessa del cluster deve essere in esecuzione ed il cluster deve essere quorato.

Per default i volumi logici creati con CLVM su di uno storage condiviso sono visibili a tutti i computer che possiedono un accesso al suddetto storage. Tuttavia sarà possibile anche creare gruppi di volumi nei quali tutti i dispositivi di storage sono visibili solo ad un nodo all'interno del cluster. È altresì possibile modificare lo stato di un gruppo di volumi da gruppo di volumi logico a gruppo di volumi clusterizzato. Per informazioni consultare [Sezione 4.3.2, «Creazione dei gruppi di volumi in un cluster»](#) e [Sezione 4.3.7, «Modifica dei parametri di un gruppo di volumi»](#).



AVVERTIMENTO

Durante la creazione dei gruppi di volumi con CLVM sullo storage condiviso, assicurarsi che tutti i nodi presenti in un cluster abbiano accesso ai volumi fisici che costituiscono il gruppo di volumi. Configurazioni cluster asimmetriche nelle quali alcuni nodi hanno un accesso allo storage non sono supportate.

Figura 1.2, «Panoramica di CLVM» mostra una panoramica di CLVM in un cluster.

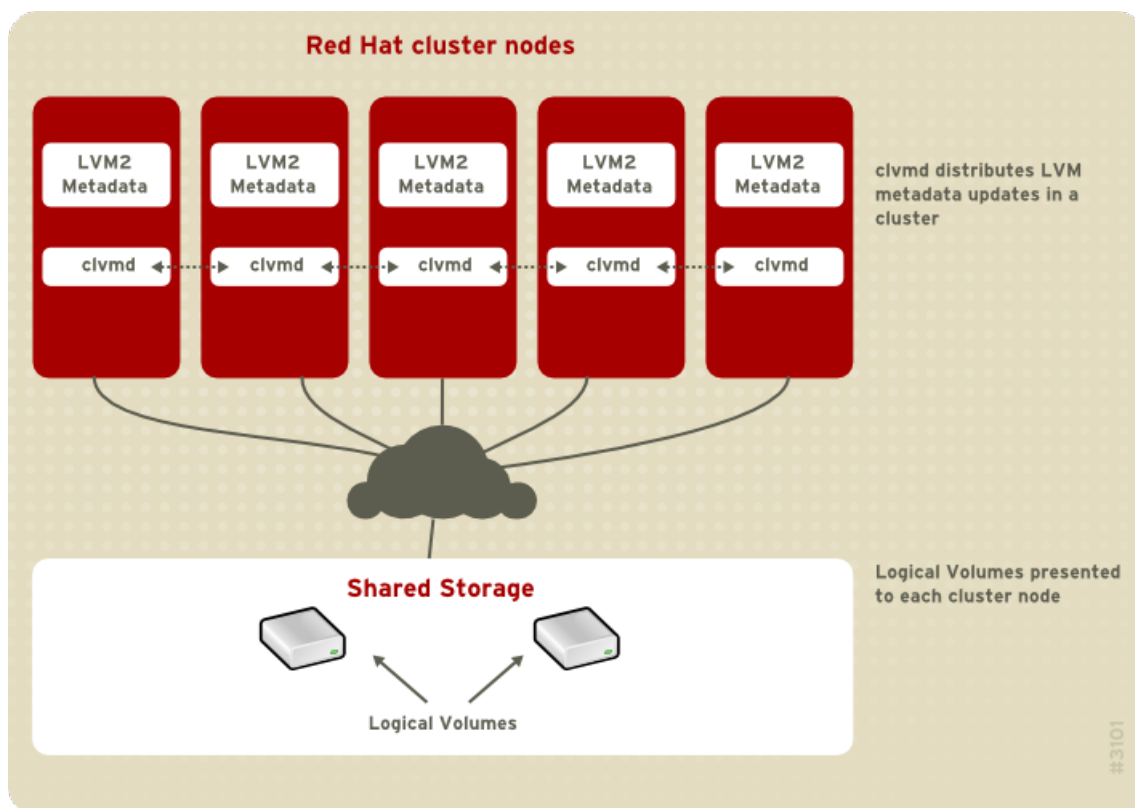


Figura 1.2. Panoramica di CLVM



NOTA

CLVM necessita di alcune modifiche al file `lvm.conf` per il blocco dell'intero cluster. Le informazioni su come configurare il file `lvm.conf` per il supporto del blocco clusterizzato sono disponibili all'interno dello stesso file `lvm.conf`. Per informazioni sul file `lvm.conf` consultare [Appendice B, File di configurazione LVM](#).

1.5. PANORAMICA DEL DOCUMENTO

Il remainder di questo documento include i seguenti capitoli:

- [Capitolo 2, Componenti di LVM](#) descrive i componenti che costituiscono un volume logico LVM.

- [Capitolo 3, *Panoramica sull'amministrazione di LVM*](#) fornisce una panoramica sulle fasi di base da seguire per la configurazione dei volumi logici LVM, sia se state usando i comandi Command Line Interface (CLI) di LVM che la Graphical User Interface (GUI).
- [Capitolo 4, *Amministrazione di LVM con i comandi CL*](#) riassume i compiti amministrativi da eseguire con i comandi CLI di LVM, per la creazione ed il mantenimento dei volumi logici.
- [Capitolo 5, *Esempi di configurazione LVM*](#) fornisce una varietà di esempi relativi alla configurazione di LVM.
- [Capitolo 6, *Troubleshooting di LVM*](#) fornisce le informazioni per il troubleshooting per una varietà di problematiche relative a LVM.
- [Capitolo 7, *Amministrazione di LVM con la GUI di LVM*](#) riassume il funzionamento della GUI di LVM.
- [Appendice A, *Device Mapper*](#) descrive il Device Mapper usato da LVM per mappare i volumi fisici e logici.
- [Appendice B, *File di configurazione LVM*](#) descrive i file di configurazione di LVM.
- [Appendice C, *Tag oggetto LVM*](#) descrive i tag dell'oggetto LVM e dell'host.
- [Appendice D, *Metadati del gruppo di volumi LVM*](#) descrive i metadati del gruppo di volumi LVM, ed include una copia dei metadati per un gruppo di volumi LVM.

CAPITOLO 2. COMPONENTI DI LVM

Questo capitolo descrive i componenti di un volume logico LVM.

2.1. VOLUMI FISICI

L'unità di base dello storage fisico di un volume logico LVM, è rappresentato da un dispositivo a blocchi come ad esempio una partizione o un intero disco. Per utilizzare il dispositivo per un volume logico LVM, è necessario inizializzare il dispositivo come physical volume 'volume fisico' (PV). Con l'inizializzazione di un dispositivo a blocchi come volume fisico, verrà inserita una etichetta vicino alla sezione iniziale del dispositivo.

Per default l'etichetta LVM viene inserita nel secondo settore di 512-byte. È possibile sovrascrivere l'impostazione predefinita inserendo l'etichetta in uno qualsiasi dei quattro settori. Ciò permette ai volumi LVM di coesistere con altri utenti dei settori interessati se necessario.

Una etichetta LVM fornisce una identificazione corretta ed un ordine preciso dei dispositivi per un dispositivo fisico, poichè i dispositivi stessi possono apparire seguendo un ordine non prestabilito durante l'avvio del sistema. Una etichetta LVM rimane costante attraverso i diversi processi di riavvio e attraverso l'intero cluster.

L'etichetta LVM identifica il dispositivo come volume fisico LVM. Presenta un identificatore unico randomico (UUID) per il volume fisico, la misura in byte del dispositivo a blocchi, e le informazioni relative alla posizione sul dispositivo dove verranno archiviati i metadati.

I metadati LVM contengono le informazioni sulla configurazione dei gruppi di volumi LVM sul sistema. Per default viene mantenuta una copia di metadati in ogni area di ogni volume fisico all'interno del gruppo di volumi. I metadati sono piccoli ed archiviati come ASCII.

Attualmente LVM permette di conservare 0, 1 o 2 copie identiche dei propri metadati su ogni volume fisico. Il valore di default è una copia. Una volta configurate le copie di metadati sul volume fisico, non sarete più in grado di modificare quel valore. La prima copia viene conservata all'inizio del dispositivo, subito dopo l'etichetta. Se è presente una seconda copia, essa viene posizionata alla fine del dispositivo. Se sovrascrivete accidentalmente l'area all'inizio del vostro disco, scrivendo su di un disco diverso da quello desiderato, una seconda copia di metadati alla fine del dispositivo vi permetterà di recuperare i metadati sovrascritti.

Per informazioni più dettagliate sui metadati LVM e sulla modifica dei parametri, consultate [Appendice D, Metadati del gruppo di volumi LVM](#)

2.1.1. Disposizione del volume fisico LVM

[Figura 2.1, «Disposizione del volume fisico»](#) mostra la disposizione di un volume fisico LVM. L'etichetta LVM è sul secondo settore, seguita dall'area dei metadati, seguita a sua volta dallo spazio utilizzabile sul dispositivo.



NOTA

Nel kernel di Linux (e su questo documento), i settori hanno una dimensione di 512 byte.

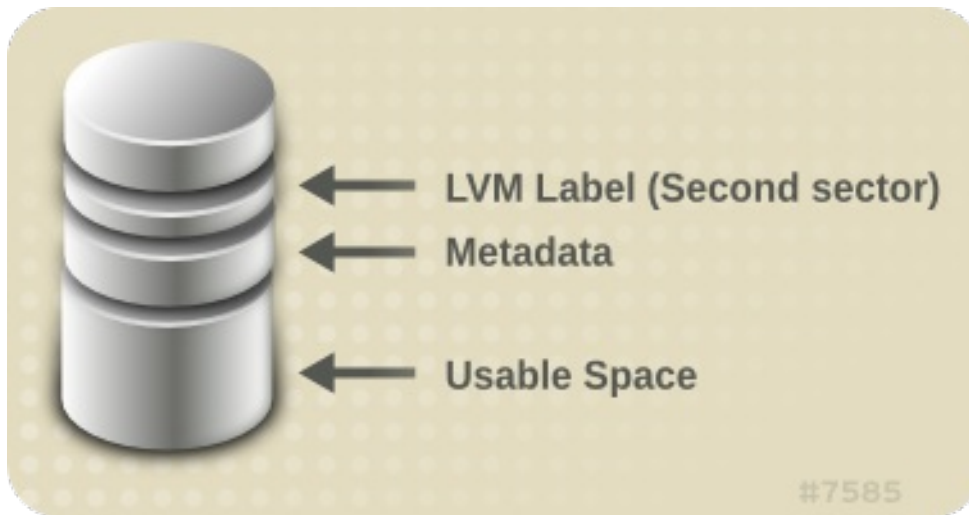


Figura 2.1. Disposizione del volume fisico

2.1.2. Partizioni multiple su di un disco

LVM permette di creare i volumi fisici dalle partizioni del disco. Generalmente è consigliato creare una partizione fisica in grado di ricoprire l'intero disco da etichettare come volume fisico LVM per i seguenti motivi:

- Amministrazione più conveniente

È più semplice controllare l'hardware in un sistema se ogni disco reale appare solo una volta. Tale tendenza è particolarmente vera se un disco fallisce. In aggiunta, i volumi fisici multipli su di un disco singolo potrebbero causare un avvertimento da parte kernel sulle diverse partizioni sconosciute al momento dell'avvio.

- Prestazioni relative allo striping

LVM non è in grado di indicare la presenza di due volumi fisici sullo stesso disco fisico. Per questo motivo se create un volume logico segmentato quando due volumi fisici si trovano sullo stesso disco fisico, i segmenti più piccoli possono essere presenti su diverse partizioni dello stesso disco. Tale comportamento potrebbe risultare in una diminuzione delle prestazioni e non in un aumento.

Potrebbero verificarsi situazioni specifiche in cui sarà necessario dividere un disco in volumi fisici LVM separati, generalmente questo processo non è consigliato. Per esempio, su di un sistema con pochi dischi potrebbe essere necessario spostare i dati sulle partizioni, durante la migrazione di un sistema esistente su volumi LVM. In aggiunta se siete in possesso di un disco molto grande e desiderate avere più di un gruppo di volumi per scopi amministrativi, allora sarà necessario partizionare il disco. Se siete in possesso di un disco con più di una partizione, e se le partizioni si trovano nello stesso gruppo di volumi, specificate quale partizione da includere in un volume logico durante la creazione di volumi segmentati.

2.2. GRUPPI DI VOLUMI

I volumi fisici vengono combinati tra loro in modo da formare i gruppi di volumi 'volume group' (VG). Ciò crea un gruppo, o pool, per lo spazio del disco dal quale i volumi logici possono essere assegnati.

All'interno del gruppo di volumi lo spazio del disco disponibile per l'assegnazione è suddiviso in unità con dimensioni fisse chiamate estensioni. Una estensione è l'unità più piccola di spazio assegnabile. All'interno del volume fisico le estensioni vengono chiamate estensioni fisiche.

Un volume logico viene assegnato nelle estensioni logiche con dimensioni identiche alle estensioni fisiche. La dimensione della estensione è così uguale per tutti i volumi logici nel gruppo di volumi. Il gruppo di volumi mappa le estensioni logiche su estensioni fisiche.

2.3. VOLUMI LOGICI LVM

In LVM un gruppo di volumi viene diviso in volumi logici. Sono disponibili tre tipi di volumi logici LVM: volumi *lineari*, volumi *segmentati* e volumi *speculari* o 'mirrored'. Essi sono descritti nelle seguenti sezioni.

2.3.1. Volumi lineari

Un volume lineare aggrega volumi fisici multipli in un volume logico. Per esempio, se siete in possesso di due dischi da 60GB, sarete in grado di creare un volume logico di 120GB. Lo storage fisico è concatenato.

Il processo di creazione di un volume lineare assegna una gamma di estensioni fisiche ad un'area di un volume logico seguendo un certo ordine. Per esempio, come mostrato in [Figura 2.2, «Mappatura estensione»](#) le estensioni logiche da 1 a 99 possono essere mappate su un volume fisico, e le estensioni logiche da 100 a 198 possono essere mappate su di un secondo volume fisico. Dal punto di vista di un'applicazione, è presente un dispositivo con dimensioni pari a 198 estensioni.

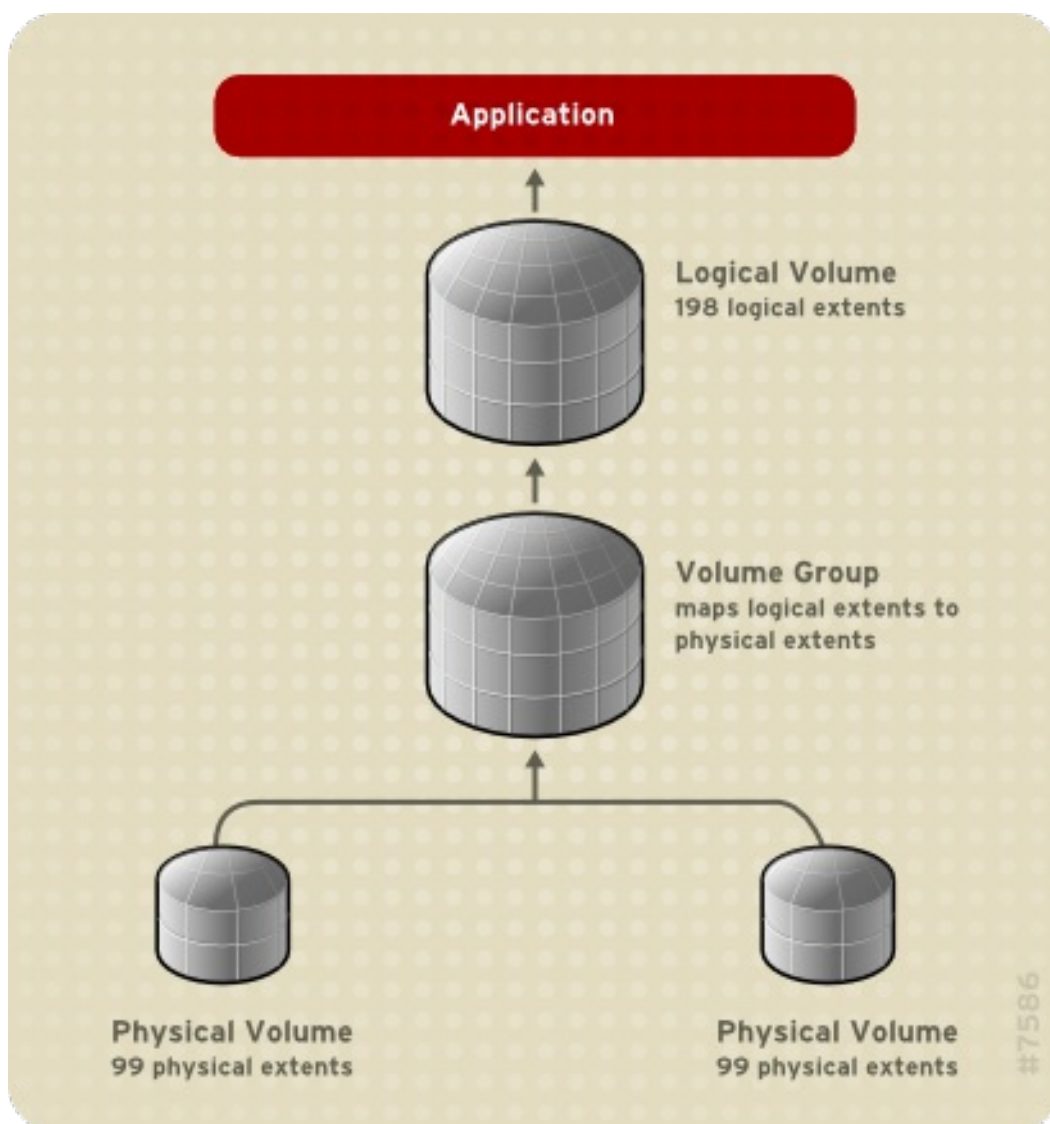


Figura 2.2. Mappatura estensione

Non è necessario che i volumi fisici che costituiscono un volume logico abbiano la stessa dimensione. [Figura 2.3, «Volume lineare con volumi fisici diversi»](#) mostra il gruppo di volumi **VG1** con una estensione fisica di dimensione pari a 4MB. Questo gruppo di volumi include 2 volumi fisici chiamati **PV1** e **PV2**. I volumi fisici sono divisi in unità di 4MB poichè questa è la dimensione dell'estensione. In questo esempio **PV1** ha una dimensione di 100 estensioni (400MB), e **PV2** una dimensione di 200 estensioni (800MB). È possibile creare un volume lineare con una dimensione che vada da 1 a 300 estensioni (da 4MB a 1200MB). In questo esempio il volume lineare chiamato **LV1** ha una dimensione di 300 estensioni.

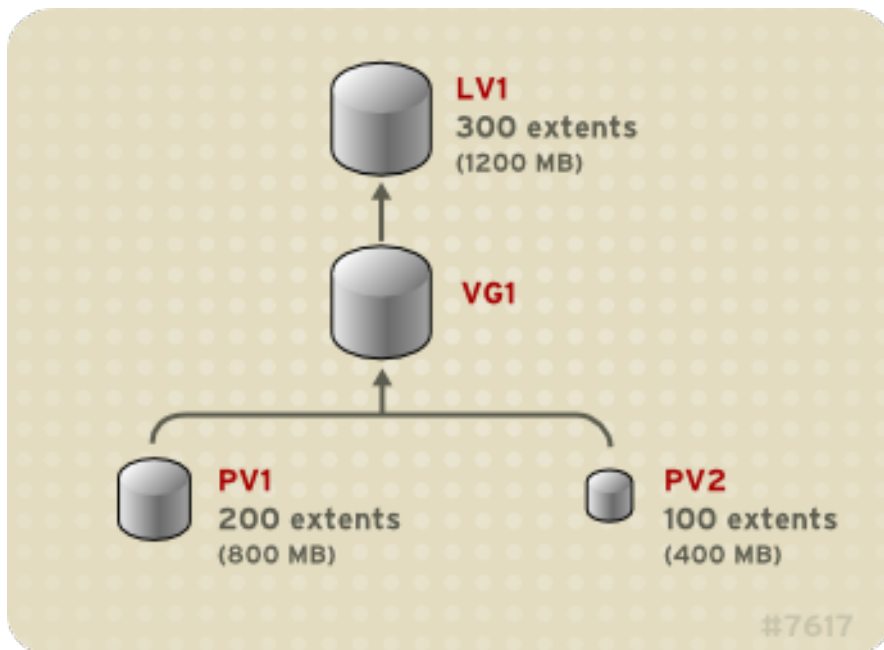


Figura 2.3. Volume lineare con volumi fisici diversi

È possibile configurare più di un volume logico lineare con una dimensione desiderata dal gruppo di estensioni fisiche. La [Figura 2.4, «Multiple Logical Volumes»](#) mostra lo stesso gruppo di volumi della [Figura 2.3, «Volume lineare con volumi fisici diversi»](#), ma in questo caso sono stati ottenuti due volumi logici dal gruppo di volumi: **LV1**, con una dimensione di 250 estensioni (1000MB) e **LV2** con una dimensione di 50 estensioni (200MB).

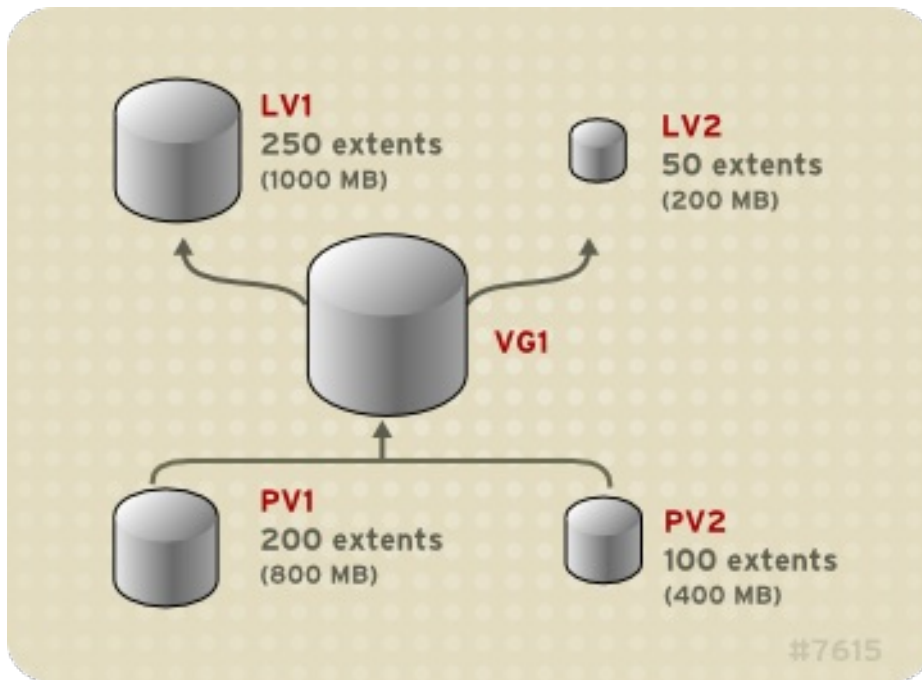


Figura 2.4. Multiple Logical Volumes

2.3.2. Volumi logici segmentati

Durante la scrittura dei dati su di un volume logico LVM il file system rilascia i dati sui volumi fisici interessati. È possibile controllare il modo attraverso il quale vengono scritti i dati sui volumi fisici, attraverso la creazione di un volume logico segmentato. Per processi di scrittura e lettura sequenziali molto grandi ciò potrebbe migliorare l'efficienza dell'I/O dei dati.

Lo strapping migliora le prestazioni attraverso la scrittura dei dati su di un numero predeterminato di volumi fisici seguendo un ordine round-robin. Con lo strapping, è possibile eseguire l'I/O in parallelo. In alcune situazioni tale comportamento può risultare in un aumento della prestazione quasi-lineare per ogni volume fisico aggiuntivo all'interno del segmento.

Il seguente mostra la scrittura dei dati attraverso tre volumi fisici. In questa figura:

- il primo segmento di dati viene scritto su PV1
- il secondo segmento viene scritto su PV2
- la terza viene scritta su PV3
- il quarto segmento viene scritto su PV1

In un volume logico segmentato la dimensione del segmento non può eccedere la dimensione di una estensione.

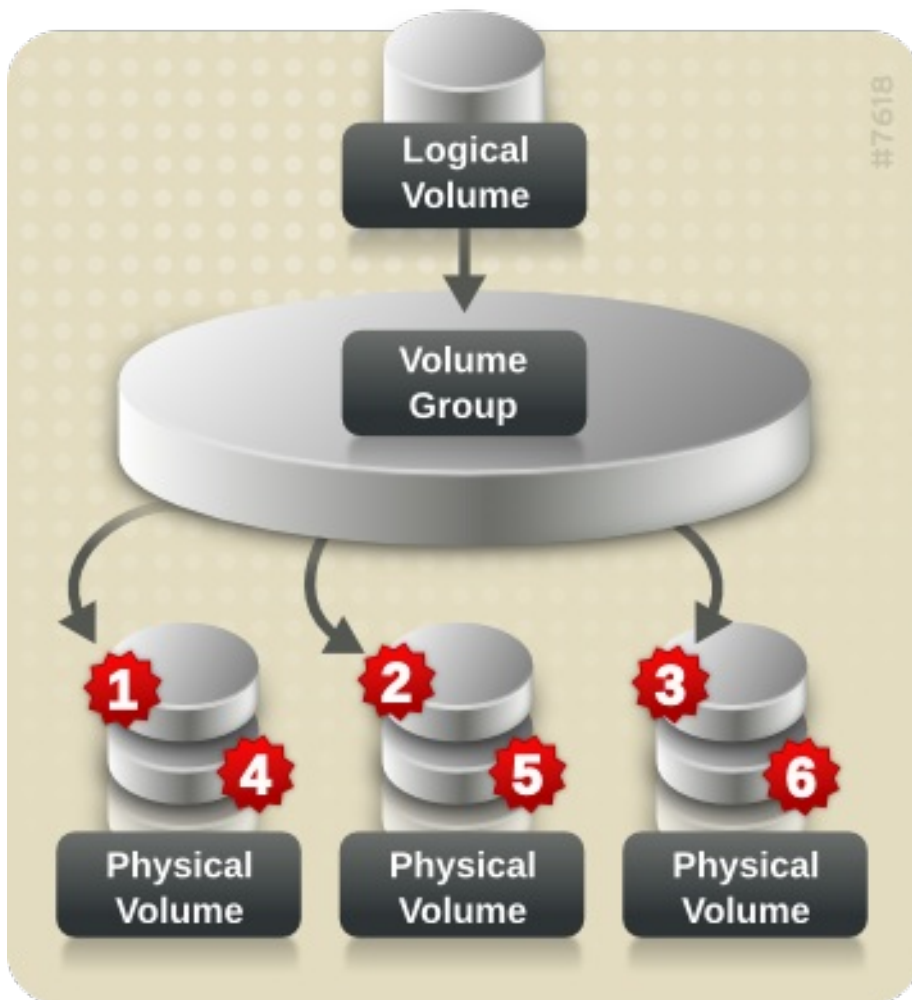


Figura 2.5. Processo di stripping dei dati su tre PV

I volumi logici segmentati possono essere estesi concatenando un altro set di dispositivi alla fine del primo set. Per poter estendere un volume logico segmentato, tuttavia, è necessario avere spazio libero sufficiente sui volumi fisici interessati che costituiscono un gruppo di volumi per supportare il segmento. Per esempio, se avete un segmento a due-vie che utilizza l'intero gruppo di volumi, l'aggiunta di un volume fisico al gruppo di volumi non vi permetterà di estendere il segmento. Al contrario, è necessario aggiungere almeno due volumi fisici al gruppo di volumi. Per maggiori informazioni su come estendere un volume segmentato consultate [Sezione 4.4.12.1, «Come estendere un volume segmentato»](#).

2.3.3. Volumi logici speculari

Un mirror mantiene copie identiche di dati su dispositivi diversi. Quando i dati vengono scritti su di un dispositivo, essi vengono scritti anche su di un secondo dispositivo rendendoli così speculari. Tale procedura garantisce una certa protezione nei confronti di un eventuale errore. Quando una sezione del mirror fallisce, il volume logico diventa volume lineare e resta così accessibile.

LVM supporta i volumi speculari. Quando create un volume logico speculare, LVM assicura che i dati scritti su di un volume fisico vengono copiati su di un volume fisico separato. Con LVM è possibile creare logical volume speculari con mirror multipli.

Un mirror LVM divide il dispositivo copiato in regioni con una misura tipica di 512KB. LVM mantiene un log piccolo il quale viene usato per mantenere un controllo sulle regioni in sincronizzazione con i mirror. Il suddetto log può essere conservato sul disco e mantenuto costante attraverso i diversi processi di riavvio, oppure può essere conservato all'interno della memoria.

Figura 2.6, «Volume logico speculare» mostra un volume logico speculare con un mirror. In questa configurazione il log viene conservato sul disco.

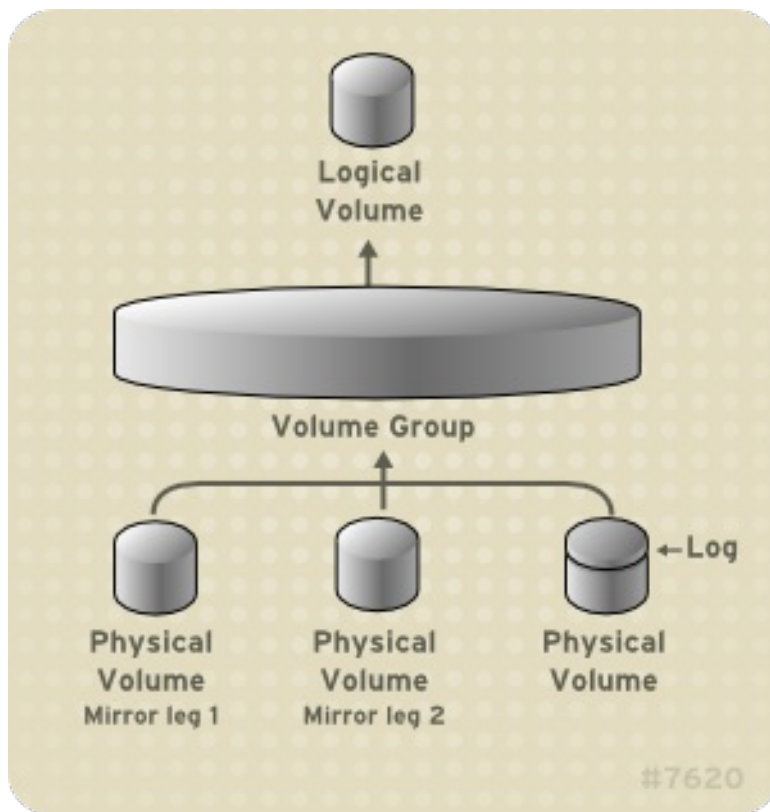


Figura 2.6. Volume logico speculare

Per informazioni su come creare e modificare i mirror consultate la [Sezione 4.4.3, «Creazione volumi speculari»](#).

2.3.4. Volumi delle snapshot

La funzione relativa alla snapshot LVM fornisce la possibilità di creare delle immagini virtuali di un dispositivo in un determinato istante senza causare l'interruzione del servizio. Quando si verifica un cambiamento del dispositivo originale dopo aver eseguito una snapshot, la funzione relativa crea una copia dell'area dei dati modificata prima dell'avvenuta modifica in modo da poter ricostruire lo stato del dispositivo.



NOTA

Le snapshot di LVM non sono supportate sui nodi di un cluster. Non sarà possibile creare un volume della snapshot in un gruppo di volumi clusterizzato.



NOTA

Le snapshot LVM non sono supportate per i volumi logici speculari LVM.

Poichè una snapshot copia solo le aree dei dati modificate dopo la creazione della snapshot stessa, la funzione relativa richiede una quantità minima di storage. Per esempio, con una origine raramente aggiornata, il 3-5% della capacità relativa è sufficiente per mantenere una snapshot.



NOTA

Le copie delle snapshot di un file system sono copie virtuali, e non media di backup per un file system. Le snapshot non forniscono un sostituto per una procedura di backup.

La dimensione della snapshot governa la quantità di spazio da usare per l'archiviazione delle modifiche relative al volume d'origine. Per esempio, se avete creato una snapshot e successivamente sovrascritto completamente l'origine, per poter archiviare le modifiche la snapshot dovrà avere una dimensione simile al volume d'origine. In tal caso modificare la snapshot con una dimensione idonea alla quantità di modifiche da apportare. Per esempio, una snapshot con un volume per la maggior parte di lettura come `/usr`, avrà bisogno di meno spazio rispetto ad una snapshot con volume sul quale vengono eseguiti un numero più elevato di processi di scrittura, come ad esempio `/home`.

Se una snapshot è eseguita piena essa sarà invalida poichè non sarà più in grado di controllare le modifiche sul volume d'origine. Le snapshot sono completamente ridimensionabili tuttavia se siete in possesso di uno spazio di storage sufficiente, sarà possibile aumentare la dimensione del volume della snapshot per evitare un suo eventuale rilascio. Al contrario, se credete che il volume della snapshot in questione risulti essere troppo grande rispetto alle vostre necessità, sarà possibile ridurre la dimensione del volume in modo da liberare lo spazio necessario per altri volumi logici.

Quando create un file system per la snapshot sarà possibile avere un accesso completo di lettura e scrittura nei confronti dell'origine. Se un segmento della snapshot viene modificato, esso viene contrassegnato e non verrà mai copiato dal volume originale.

È possibile utilizzare la funzione delle snapshot in diversi modi:

- È consigliato fare una istantanea se è necessario eseguire un backup su di un volume logico senza arrestare l'aggiornamento 'live' dei dati da parte del sistema.
- È possibile eseguire il comando `fsck` sul file system di una snapshot per controllare l'integrità del file system e determinare se il file system originale ha bisogno di correzioni.
- Poichè è possibile eseguire la lettura e scrittura della snapshot, è possibile testare le applicazioni con i dati di produzione eseguendo una snapshot e successivamente alcuni test senza interessare i dati reali.
- È possibile altresì creare i volumi LVM per un loro utilizzo con tecnologie di virtualizzazione Red Hat. Le snapshot LVM possono essere usate per creare snapshot di immagini del guest virtuale. Le suddette snapshot sono in grado di fornire un metodo conveniente per la modifica dei guest esistenti o per la creazione di nuovi guest con storage aggiuntivo minimo. Per maggiori informazioni sulla creazione di snapshot LVM per guest virtualizzati consultare la *Red Hat Enterprise Linux Virtualization Guide*.

Per informazioni su come creare i volumi della snapshot consultare [Sezione 4.4.4, «Creazione dei volumi della snapshot»](#).

Con la release di Red Hat Enterprise Linux 6 sarà possibile usare l'opzione `- -merge` del comando `lvconvert` per eseguire il merge di una snapshot nel proprio volume d'origine. Questa funzione può essere usata se si desidera eseguire un rollback del sistema in presenza di perdita di dati o file o se avete la necessità di ripristinare il sistema ad uno stato precedente. Dopo il merge del volume della snapshot, il volume logico risultante avrà il nome del volume d'origine, il numero minore, l'UUID con una conseguente rimozione della snapshot sulla quale è stato eseguito il merge. Per informazioni su come usare questa opzione consultare [Sezione 4.4.5, «Unione dei volumi della snapshot»](#).

CAPITOLO 3. PANORAMICA SULL'AMMINISTRAZIONE DI LVM

Questo capitolo fornisce una panoramica sulle procedure amministrative usate per la configurazione dei volumi logici LVM. Esso fornisce un quadro generale delle fasi interessate. Per esempi specifici passo-dopo-passo di procedure comuni per la configurazione di LVM, consultate [Capitolo 5, Esempi di configurazione LVM](#).

Per una descrizione dei comandi CLI utilizzati per gestire LVM consultate [Capitolo 4, Amministrazione di LVM con i comandi CLI](#). Alternativamente potrete utilizzare la GUI di LVM descritta in [Capitolo 7, Amministrazione di LVM con la GUI di LVM](#).

3.1. CREAZIONE DEI VOLUMI LVM IN UN CLUSTER

Per creare i volumi logici in un ambiente cluster utilizzare il Clustered Logical Volume Manager (CLVM), il quale è costituito da un insieme di estensioni per LVM. Le suddette estensioni permettono ad un cluster di computer di gestire lo storage condiviso (per esempio su di un SAN) utilizzando LVM. Per poter utilizzare CLVM il software Red Hat Cluster Suite, incluso il demone `clmvd`, deve essere avviato al momento dell'avvio come riportato in [Sezione 1.4, «LVM Logical Volume Manager \(CLVM\)»](#).

Il processo di creazione dei volumi logici di LVM in un ambiente cluster è identico alla creazione dei volumi logici LVM in un nodo singolo. Non vi è alcuna differenza nei comandi LVM utilizzati o nell'interfaccia GUI di LVM. Per poter abilitare i volumi di LVM che state creando in un cluster, l'infrastruttura del cluster deve essere in esecuzione e possedere un numero sufficiente di membri (quorate).

CLVM necessita di alcune modifiche al file `lvm.conf` per il blocco dell'intero cluster. Le informazioni su come configurare il file `lvm.conf` per il supporto del blocco clusterizzato sono disponibili all'interno dello stesso file `lvm.conf`. Per informazioni sul file `lvm.conf` consultare [Appendice B, File di configurazione LVM](#).

Per default i volumi logici creati con CLVM su di uno storage condiviso sono visibili a tutti i computer che possiedono un accesso allo storage condiviso. Tuttavia, sarà possibile anche creare i volumi logici quando i dispositivi di storage sono visibili solo ad un nodo all'interno del cluster. È altresì possibile modificare lo stato di un gruppo di volumi da un gruppo di volumi locali ad uno clusterizzato. Per informazioni consultare [Sezione 4.3.2, «Creazione dei gruppi di volumi in un cluster»](#) e [Sezione 4.3.7, «Modifica dei parametri di un gruppo di volumi»](#).



AVVERTIMENTO

Durante la creazione dei gruppi di volumi con CLVM sullo storage condiviso, assicurarsi che tutti i nodi presenti in un cluster abbiano accesso ai volumi fisici che costituiscono il gruppo di volumi. Configurazioni cluster asimmetriche nelle quali alcuni nodi hanno un accesso allo storage non sono supportate.

Per informazioni su come installare High Availability Add-On ed impostare l'infrastruttura del cluster consultate [Amministrazione del Cluster](#).

Per un esempio su come creare un volume logico speculare in un cluster consultare [Sezione 5.5, «Creazione di un volume logico LVM speculare in un cluster»](#).

3.2. PANORAMICA SULLA CREAZIONE DEL VOLUME LOGICO

Il seguente è un sommario delle fasi necessarie per eseguire la creazione di un volume logico LVM.

1. Inizializzare le partizioni da utilizzare per il volume LVM come volume fisico (tale operazione aggiungerà loro una etichetta).
2. Creazione di un gruppo di volumi
3. Creazione di un volume logico.

Dopo aver creato il volume logico sarà possibile creare e montare il file system. Gli esempi in questo documento usano i file system GFS2.



NOTA

Anche se un file system GFS2 può essere implementato in un sistema standalone o come parte di una configurazione cluster per la release Red Hat Enterprise 6, Red Hat non supporta l'utilizzo di GFS2 come file system con nodo singolo. Red Hat continuerà a supportare file system GFS2 con nodo singolo per il montaggio delle snapshot dei file system del cluster (per esempio per l'esecuzione dei backup).

1. Create un file system GFS2 sul volume logico con il comando `mkfs.gfs2`.
2. Create un nuovo mount point con il comando `mkdir`. In un sistema clusterizzato, create il mount point su tutti i nodi nel cluster.
3. Montate il file system. Potreste aggiungere una riga a `fstab` per ogni nodo nel sistema.

Alternativamente potrete creare e montare il file system GFS2 con la GUI di LVM.

La creazione del volume LVM è indipendente dalla macchina poichè l'area di storage per le informazioni sull'impostazione di LVM risulta essere sui volumi fisici e non sulla macchina sulla quale è stato creato il volume. I server che utilizzano lo storage presentano copie locali ma possono eseguire una riproduzione in base al contenuto dei volumi fisici. È possibile collegare i volumi fisici su di un server diverso se le versioni di LVM sono compatibili.

3.3. SVILUPPO DI UN FILE SYSTEM SU DI UN VOLUME LOGICO

Per sviluppare un file system su di un volume logico, seguite le seguenti fasi:

1. Create un nuovo volume fisico.
2. Estendete il gruppo di volumi che contiene il volume logico con il file system che state sviluppando in modo da includere il nuovo volume fisico.
3. Aumentate il volume logico in modo da includere il nuovo volume fisico.
4. Sviluppo del file system.

Se siete in possesso di spazio sufficiente non assegnato nel gruppo dei volumi, allora potrete utilizzare il suddetto spazio per estendere il volume logico invece di eseguire le fasi 1 e 2.

3.4. BACKUP DEL VOLUME LOGICO

I backup dei metadati e gli archivi vengono creati automaticamente ad ogni cambiamento della configurazione del volume logico e del gruppo di volumi se non precedentemente disabilitato nel file `lvm.conf`. Per default, il backup dei metadati viene conservato in `/etc/lvm/backup` mentre gli archivi vengono conservati in `/etc/lvm/archive`. La durata dell'archiviazione degli archivi dei metadati in `/etc/lvm/archive` ed il numero dei file d'archivio è determinato dai parametri impostati nel file `lvm.conf`. Un backup giornaliero del sistema dovrebbe includere i contenuti della directory `/etc/lvm` nel backup stesso.

Da notare che il backup dei metadati non esegue il backup dei dati del sistema e dell'utente contenuti nei volumi logici.

È possibile eseguire il backup manuale dei metadati sul file `/etc/lvm/backup` con il comando `vgcfgbackup`. È possibile ripristinare i metadati con il comando `vgcfgrestore`. I comandi `vgcfgbackup` e `vgcfgrestore` sono descritti in [Sezione 4.3.12, «Esecuzione del back up dei metadati del gruppo di volumi»](#).

3.5. REGISTRAZIONE

L'output del messaggio passa attraverso un modulo di registrazione con scelte indipendenti dei livelli di registrazione per:

- output/errore standard
- syslog
- file di log
- funzione di log esterna

I livelli di registrazione sono impostati in `/etc/lvm/lvm.conf`, il quale viene descritto in [Appendice B, *File di configurazione LVM*](#).

CAPITOLO 4. AMMINISTRAZIONE DI LVM CON I COMANDI CLI

Questo capitolo riassume i compiti amministrativi individuali che si possono eseguire tramite i comandi della Command Line Interface (CLI) di LVM, per creare e gestire i volumi logici.



NOTA

Se state creando o modificando un volume LVM per un ambiente clusterizzato assicuratevi che il demone `clvmd` sia in esecuzione. Per maggiori informazioni consultare [Sezione 3.1, «Creazione dei volumi LVM in un cluster»](#).

4.1. COME UTILIZZARE I COMANDI CLI

Sono presenti numerose funzioni generali relative al CLI di LVM.

Se è necessario specificare delle dimensioni in un argomento della linea di comando sarà possibile specificare le unità in modo esplicito. Se non viene specificata alcuna unità allora verrà assunto un valore predefinito, generalmente KB o MB. Il CLI di LVM non accetta frazioni.

Quando specificate le unità in un argomento della linea di comando, LVM non distingue le lettere minuscole da quelle maiuscole; per esempio non vi è alcuna differenza tra M e m, ed utilizza una potenza di due 2 (multiplo di 1024). Tuttavia quando specificate l'opzione `--units` in un comando, la lettere minuscole indicano che le unità sono multipli di 1024, mentre lettere maiuscole indicano che le stesse unità sono multiple di 1000.

Dove i comandi accettano il gruppo di volumi o i nomi del volume logico come argomenti, il nome completo del percorso è facoltativo. Un volume logico chiamato `lvol10` in un gruppo di volumi `vg0` può essere specificato come `vg0/lvol10`. Dove un elenco dei gruppi di volumi è necessario ma lasciato vuoto verrà visualizzato un elenco di tutti i gruppi di volumi. Dove sarà necessario un elenco di volumi logici ma al contrario viene fornito un gruppo di volumi, verrà visualizzato un elenco di tutti i volumi logici per quel gruppo di volumi. Per esempio il comando `lvdisplay vg0` visualizzerà tutti i volumi logici nel gruppo di volumi `vg0`.

Tutti i comandi di LVM accettano un argomento `-v` il quale può essere inserito numerose volte per aumentare la verbosità di un output. Per esempio, di seguito vengono mostrati gli esempi relativi all'output predefinito del comando `lvcreate`.

```
# lvcreate -L 50MB new_vg
Rounding up size to full physical extent 52.00 MB
Logical volume "lvol10" created
```

Il seguente comando mostra l'output di `lvcreate` con l'argomento `-v`.

```
# lvcreate -v -L 50MB new_vg
Finding volume group "new_vg"
Rounding up size to full physical extent 52.00 MB
Archiving volume group "new_vg" metadata (seqno 4).
Creating logical volume lvol10
Creating volume group backup "/etc/lvm/backup/new_vg" (seqno 5).
Found volume group "new_vg"
Creating new_vg-lvol10
Loading new_vg-lvol10 table
Resuming new_vg-lvol10 (253:2)
```

```

    Clearing start of logical volume "lvol0"
    Creating volume group backup "/etc/lvm/backup/new_vg" (seqno 5).
    Logical volume "lvol0" created

```

È possibile usare anche gli argomenti `-vv`, `-vvv` o the `-vvvv` per visualizzare informazioni più dettagliate sull'esecuzione del comando. L'argomento `-vvvv` fornisce la quantità massima di informazioni in questo momento. Il seguente esempio mostra solo le prime righe dell'output per il comando `lvcreate` con l'argomento `-vvvv`.

```

# lvcreate -vvvv -L 50MB new_vg
#lvmcmdline.c:913      Processing: lvcreate -vvvv -L 50MB new_vg
#lvmcmdline.c:916      O_DIRECT will be used
#config/config.c:864   Setting global/locking_type to 1
#locking/locking.c:138 File-based locking selected.
#config/config.c:841   Setting global/locking_dir to /var/lock/lvm
#activate/activate.c:358 Getting target version for linear
#ioctl/libdm-iface.c:1569 dm version    OF    [16384]
#ioctl/libdm-iface.c:1569 dm versions   OF    [16384]
#activate/activate.c:358 Getting target version for striped
#ioctl/libdm-iface.c:1569 dm versions   OF    [16384]
#config/config.c:864   Setting activation/mirror_region_size to 512
...

```

È possibile visualizzare l'aiuto per qualsiasi dei comandi CLI di LVM con l'argomento `--help` del comando.

```
commandname --help
```

Per visualizzare la pagina man di un comando, eseguite il comando `man`:

```
man commandname
```

Il comando `man lvm` fornisce informazioni generali online su LVM.

Tutti gli oggetti LVM vengono indicati internamente con un UUID, il quale viene assegnato al momento della creazione dell'oggetto. Ciò può essere utile in una situazione dove verrà rimosso un volume fisico chiamato `/dev/sdf`, facente parte di un gruppo di volumi e, quando nuovamente collegato, chiamato `/dev/sdk`. LVM sarà in grado di trovare il volume fisico poiché è in grado di identificare l'UUID corrispondente e non il nome del dispositivo. Per informazioni su come specificare l'UUID di un volume fisico durante la creazione di un volume fisico, consultate la [Sezione 6.4, «Recupero dei metadati del volume fisico»](#).

4.2. AMMINISTRAZIONE DEL VOLUME FISICO

Questa sezione descrive i comandi che eseguono le diverse funzioni per l'amministrazione del volume fisico.

4.2.1. Creazione dei volumi fisici

Le sottosezioni seguenti descrivono i comandi usati per la creazione di volumi fisici.

4.2.1.1. Impostazione del tipo di partizione

Se state utilizzando l'intero dispositivo a disco per il vostro volume fisico, il disco non deve avere alcuna tabella delle partizioni. Per partizioni del disco DOS, l'id della partizione dovrebbe essere impostata su 0x8e usando `fdisk` o `cfdisk` o equivalente. Per tutti i dispositivi a disco solo la tabella delle partizioni deve essere rimossa. Tale processo distruggerà tutti i dati sul disco in questione. È possibile rimuovere una tabella delle partizioni esistente azzerando il primo settore con il seguente comando:

```
dd if=/dev/zero of=PhysicalVolume bs=512 count=1
```

4.2.1.2. Inizializzazione dei volumi fisici

Utilizzate il comando `pvcreate` per inizializzare un dispositivo a blocchi da usare come volume fisico. L'inizializzazione è analoga alla formattazione di un file system.

Il seguente comando inizializza `/dev/sdd1`, `/dev/sde1`, e `/dev/sdf1` per l'uso come volumi fisici LVM.

```
pvcreate /dev/sdd1 /dev/sde1 /dev/sdf1
```

Per inizializzare le partizioni e non l'intero disco: eseguire `pvcreate` sulla partizione. Nel seguente esempio `/dev/hdb1` viene inizializzato come volume fisico LVM, ed usato in un secondo momento come parte di un volume logico LVM.

```
pvcreate /dev/hdb1
```

4.2.1.3. Scansione per dispositivi a blocchi

È possibile eseguire una scansione dei dispositivi a blocchi che possono essere usati come volumi fisici con il comando `lvmdiskscan`, come riportato nel seguente esempio.

```
# lvmdiskscan
/dev/ram0          [          16.00 MB]
/dev/sda          [          17.15 GB]
/dev/root         [          13.69 GB]
/dev/ram          [          16.00 MB]
/dev/sda1         [          17.14 GB] LVM physical volume
/dev/VolGroup00/LogVol01 [        512.00 MB]
/dev/ram2         [          16.00 MB]
/dev/new_vg/lvol0 [          52.00 MB]
/dev/ram3         [          16.00 MB]
/dev/pkl_new_vg/sparkie_lv [         7.14 GB]
/dev/ram4         [          16.00 MB]
/dev/ram5         [          16.00 MB]
/dev/ram6         [          16.00 MB]
/dev/ram7         [          16.00 MB]
/dev/ram8         [          16.00 MB]
/dev/ram9         [          16.00 MB]
/dev/ram10        [          16.00 MB]
/dev/ram11        [          16.00 MB]
/dev/ram12        [          16.00 MB]
/dev/ram13        [          16.00 MB]
/dev/ram14        [          16.00 MB]
/dev/ram15        [          16.00 MB]
```

```

/dev/sdb          [      17.15 GB]
/dev/sdb1         [      17.14 GB] LVM physical volume
/dev/sdc          [      17.15 GB]
/dev/sdc1         [      17.14 GB] LVM physical volume
/dev/sdd          [      17.15 GB]
/dev/sdd1         [      17.14 GB] LVM physical volume
7 disks
17 partitions
0 LVM physical volume whole disks
4 LVM physical volumes

```

4.2.2. Visualizzazione dei volumi fisici

Per la visualizzazione delle proprietà dei volumi fisici LVM sono disponibili tre comandi: **pvs**, **pvdisplay**, e **pvscan**.

Il comando **pvs** fornisce le informazioni sul volume fisico attraverso un formato configurabile, mostrando una riga per volume fisico. Il comando **pvs** è idoneo per un controllo del formato ed è utile per lo scripting. Per informazioni su come utilizzare il comando **pvs** per personalizzare l'output desiderato consultate [Sezione 4.8, «Personalizzazione dei riporti per LVM»](#).

Il comando **pvdisplay** fornisce un output verboso con diverse righe per ogni volume fisico. Esso visualizza le proprietà fisiche (dimensione, estensioni, gruppo di volumi ecc) in un formato fisso.

Il seguente esempio mostra l'output di **pvdisplay** per un volume fisico singolo.

```

# pvdisplay
--- Physical volume ---
PV Name           /dev/sdc1
VG Name           new_vg
PV Size           17.14 GB / not usable 3.40 MB
Allocatable       yes
PE Size (KByte)   4096
Total PE          4388
Free PE           4375
Allocated PE      13
PV UUID           Joqlch-yWSj-kuEn-IdwM-01S9-X08M-mcpsVe

```

Il comando **pvscan** esegue la scansione di tutti i dispositivi a blocchi LVM nel sistema per la presenza di volumi fisici.

Il seguente comando mostra tutti i dispositivi fisici trovati:

```

# pvscan
PV /dev/sdb2   VG vg0    lvm2 [964.00 MB / 0    free]
PV /dev/sdc1   VG vg0    lvm2 [964.00 MB / 428.00 MB free]
PV /dev/sdc2   VG          lvm2 [964.84 MB]
Total: 3 [2.83 GB] / in use: 2 [1.88 GB] / in no VG: 1 [964.84 MB]

```

È possibile definire un filtro in **lvm.conf**, così facendo questo comando non esegue la scansione dei volumi fisici specifici. Per informazioni su come utilizzare i filtri per controllare su quali dispositivi eseguire la scansione consultate la [Sezione 4.5, «Controllo delle scansioni del dispositivo LVM con i filtri»](#).

4.2.3. Come impedire l'assegnazione su di un volume fisico

È possibile impedire l'assegnazione di estensioni fisiche sullo spazio disponibile di uno o più volumi fisici tramite il comando **pvchange**. Tale operazione potrebbe essere necessaria se sono presenti sul disco alcuni errori, oppure se desiderate rimuovere il volume fisico.

Il seguente comando non permette l'assegnazione di estensioni fisiche su **/dev/sdk1**.

```
pvchange -x n /dev/sdk1
```

È possibile usare anche gli argomenti **-xy** del comando **pvchange**, per abilitare l'assegnazione dove precedentemente non permessa.

4.2.4. Come variare la dimensione di un volume fisico

Se desiderate modificare la dimensione di un dispositivo a blocchi utilizzate il comando **pvresize** per aggiornare LVM con la nuova dimensione. Potrete eseguire questo comando mentre LVM utilizza il volume fisico.

4.2.5. Rimozione dei volumi fisici

Se un dispositivo non è più necessario potrete rimuovere l'etichetta LVM con il comando **pvremove**. L'esecuzione del comando **pvremove** azzererà i metadati di LVM su di un volume fisico vuoto.

Se il volume fisico che desiderate rimuovere è parte di un gruppo di volumi, allora è necessario rimuoverlo dal gruppo in questione con **vgreduce**, come descritto in [Sezione 4.3.6, «Rimozione dei volumi fisici da un gruppo di volumi»](#).

```
# pvremove /dev/ram15
Labels on physical volume "/dev/ram15" successfully wiped
```

4.3. AMMINISTRAZIONE DEL GRUPPO DI VOLUMI

Questa sezione descrive i comandi usati per eseguire i compiti di amministrazione del gruppo di volumi.

4.3.1. Creazione dei gruppi di volumi

Per creare un gruppo di volumi da uno o più volumi fisici usare il comando **vgcreate**. Il comando **vgcreate** crea un nuovo gruppo di volumi con un nome ed aggiunge ad esso almeno un volume fisico.

Il seguente comando crea un gruppo di volumi chiamato **vg1** il quale contiene i volumi fisici **/dev/sdd1** e **/dev/sde1**.

```
vgcreate vg1 /dev/sdd1 /dev/sde1
```

Quando i volumi fisici vengono usati per creare un gruppo di volumi, lo spazio del disco viene diviso per default in estensioni di 4MB. Questa è la quantità minima dalla quale la dimensione del volume logico può essere aumentata o diminuita. Un numero molto grande di estensioni non avrà alcun impatto sulle prestazioni I/O del volume logico.

Potrete specificare la dimensione dell'estensione con il comando **vgcreate** se il valore predefinito non risulta essere idoneo utilizzando l'argomento **-s**. È possibile inserire dei limiti sul numero di volumi

logici o fisici ospitati dal gruppo di volumi, utilizzando gli argomenti **-p** e **-l** del comando **vgcreate**.

Per default un gruppo di volumi assegna estensioni fisiche seguendo regole comuni, come ad esempio non posizionare i segmenti paralleli sullo stesso volume fisico. Ciò fa' parte della politica di assegnazione **normal**. È possibile utilizzare l'argomento **--alloc** del comando **vgcreate**, per specificare una policy di assegnazione di tipo **contiguous**, **anywhere**, o **cling**.

Con la politica **contiguous** le nuove estensioni sono adiacenti alle estensioni esistenti. Se è disponibile un numero sufficiente di estensioni per poter soddisfare una richiesta di assegnazione, e se la politica di assegnazione **normal** non le usa, la politica **anywhere** sarà in grado di farlo anche a rischio di ridurre la prestazione, a causa del posizionamento di due segmenti sullo stesso volume fisico. La politica **cling** posiziona le nuove estensioni sullo stesso volume fisico delle estensioni esistenti, nello stesso segmento del volume logico. Queste politiche possono essere modificate usando il comando **vgchange**.

Per informazioni su come utilizzare la politica **cling** insieme ai tag LVM per specificare i volumi aggiuntivi da usare durante l'estensione di un volume LVM consultare [Sezione 4.4.12.2, «Estensione di un volume logico con politica di assegnazione cling»](#).

In generale, le politiche per l'assegnazione diverse da **normal** sono necessarie solo in casi particolari, quando è richiesto specificare un'assegnazione di estensioni non standard o non comune.

Il gruppo di volumi ed i volumi logici sottostanti sono inclusi nell'albero della directory del file speciale del dispositivo all'interno di **/dev** con la seguente disposizione:

```
/dev/vg/lv/
```

Per esempio, se create due gruppi di volumi **myvg1** e **myvg2**, ognuno dei quali con tre logical volumes chiamati **lv01**, **lv02**, e **lv03**, ciò creerà sei file speciali del dispositivo:

```
/dev/myvg1/lv01
/dev/myvg1/lv02
/dev/myvg1/lv03
/dev/myvg2/lv01
/dev/myvg2/lv02
/dev/myvg2/lv03
```

La dimensione massima del dispositivo con LVM è 8 Exabyte su CPU a 64-bit.

4.3.2. Creazione dei gruppi di volumi in un cluster

È possibile creare i gruppi di volumi in un ambiente cluster tramite il comando **vgcreate**, in modo simile ad una loro creazione su di un nodo singolo.

Per default i gruppi di volumi creati con CLVM su di uno storage condiviso, sono visibili a tutti i computer che possiedono un accesso allo storage condiviso. Tuttavia, sarà possibile anche creare i volumi logici quando i dispositivi di storage sono visibili solo ad un nodo all'interno del cluster usando **-c n** del comando **vgcreate**.

Il seguente comando, se eseguito in un ambiente cluster, crea un gruppo di volumi locale al nodo dal quale il comando è stato eseguito. Il comando crea un volume logico chiamato **vg1** il quale contiene i volumi fisici **/dev/sdd1** e **/dev/sde1**.

```
vgcreate -c n vg1 /dev/sdd1 /dev/sde1
```

È possibile modificare un gruppo di volumi esistente se locale o clusterizzato tramite l'opzione `-c` del comando `vgchange`, come riportato in [Sezione 4.3.7, «Modifica dei parametri di un gruppo di volumi»](#).

È possibile controllare se un gruppo di volumi esistente risulta essere un gruppo di volumi clusterizzato utilizzando il comando `vgs`, il quale mostra l'attributo `c` se il volume è clusterizzato. Il seguente comando mostra gli attributi dei gruppi di volumi `VolGroup00` e `testvg1`. In questo esempio `VolGroup00` non è clusterizzato mentre `testvg1` risulta esserlo, come riportato dall'attributo `c` con l'intestazione `Attr`.

```
[root@doc-07]# vgs
VG                #PV #LV #SN Attr   VSize  VFree
VolGroup00        1   2   0 wz--n- 19.88G    0
testvg1           1   1   0 wz--nc 46.00G  8.00M
```

Per maggiori informazioni sul comando `vgs` consultare [Sezione 4.3.4, «Come visualizzare i gruppi di volumi»](#) [Sezione 4.8, «Personalizzazione dei riporti per LVM»](#), e la pagina man di `vgs`.

4.3.3. Aggiunta di un volume fisico ad un gruppo di volumi

Per aggiungere un volume fisico in un gruppo di volumi esistente, utilizzate il comando `vgextend`. Il comando `vgextend` aumenta la capacità di un gruppo di volumi aggiungendo uno o più volumi fisici disponibili.

Il seguente comando aggiunge il volume fisico `/dev/sdf1` al gruppo di volumi `vg1`.

```
vgextend vg1 /dev/sdf1
```

4.3.4. Come visualizzare i gruppi di volumi

Per poter visualizzare le proprietà dei gruppi di volumi di LVM sono disponibili i seguenti comandi: `vgs` e `vgdisplay`.

Il comando `vgscan` può essere utilizzato per visualizzare i gruppi di volumi, anche se il suo scopo primario è quello di eseguire la scansione di tutti i dischi per i gruppi di volumi e ricreare il file cache di LVM. Per informazioni sul comando `vgscan`, consultate la [Sezione 4.3.5, «Scansione dischi per i gruppi di volumi per la creazione del file di cache»](#).

Il comando `vgs` fornisce le informazioni sul gruppo di volumi attraverso una forma configurabile, mostrando una riga per gruppo. Il comando `vgs` è idoneo per un controllo del formato ed è utile per lo scripting. Per informazioni su come utilizzare il comando `vgs` per personalizzare il vostro output consultate la [Sezione 4.8, «Personalizzazione dei riporti per LVM»](#).

Il comando `vgdisplay` mostra le proprietà del gruppo di volumi (come ad esempio dimensione, estensioni, numero di volumi fisici, ecc) attraverso una forma fissa. Il seguente esempio mostra l'output di un comando `vgdisplay` per il gruppo di volumi `new_vg`. Se non specificate un gruppo di volumi, tutti i gruppi di volumi esistenti verranno visualizzati.

```
# vgdisplay new_vg
--- Volume group ---
VG Name                new_vg
System ID
Format                 lvm2
Metadata Areas         3
```

```

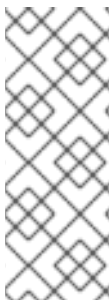
Metadata Sequence No 11
VG Access             read/write
VG Status             resizable
MAX LV               0
Cur LV              1
Open LV              0
Max PV               0
Cur PV              3
Act PV               3
VG Size              51.42 GB
PE Size              4.00 MB
Total PE             13164
Alloc PE / Size      13 / 52.00 MB
Free PE / Size       13151 / 51.37 GB
VG UUID              jxQJ0a-ZKk0-OpM0-0118-nlw0-wwqd-fD5D32

```

4.3.5. Scansione dischi per i gruppi di volumi per la creazione del file di cache

Il comando **vgscan** esegue la scansione di tutti i dispositivi a disco supportati nel sistema, andando alla ricerca dei volumi fisici LVM e dei gruppi dei volumi. Ciò crea il file di cache di LVM in `/etc/lvm/.cache`, il quale mantiene un elenco di dispositivi LVM correnti.

LVM esegue il comando **vgscan** automaticamente al momento dell'avvio del sistema ed in altri momenti durante il funzionamento di LVM, ad esempio durante l'esecuzione del comando **vgcreate**, oppure quando LVM rileva una qualsiasi inconsistenza.



NOTA

Potrebbe essere necessario eseguire manualmente il comando **vgscan** se modificate la configurazione del vostro hardware aggiungendo o rimuovendo il dispositivo da un nodo, rendendo così visibili al sistema i nuovi dispositivi i quali non erano presenti al momento dell'avvio. Tale procedura potrebbe essere necessaria, per esempio, se aggiungete nuovi dischi al sistema su di un SAN, oppure se eseguite l'hotplug di un nuovo disco etichettato come volume fisico.

All'interno del file `lvm.conf` è possibile definire un filtro per limitare la scansione, e quindi evitare dispositivi specifici. Per maggiori informazioni su come utilizzare i filtri per controllare i dispositivi sui quali eseguire la scansione, consultate la [Sezione 4.5, «Controllo delle scansioni del dispositivo LVM con i filtri»](#).

Il seguente esempio mostra l'output di un comando **vgscan**.

```

# vgscan
Reading all physical volumes. This may take a while...
Found volume group "new_vg" using metadata type lvm2
Found volume group "officevg" using metadata type lvm2

```

4.3.6. Rimozione dei volumi fisici da un gruppo di volumi

Per rimuovere i volumi fisici non utilizzati dal gruppo di volumi utilizzate il comando **vgreduce**. Il comando **vgreduce** riduce la capacità di un gruppo di volumi attraverso la rimozione di uno o più volumi fisici. Tale operazione libera i volumi fisici in modo da usarli in altri gruppi, oppure da rimuovere completamente dal sistema.

Prima di rimuovere un volume fisico da un gruppo di volumi, assicuratevi che il volume fisico non sia stato usato da qualsiasi altro volume logico tramite il comando **pvdisplay**.

```
# pvdisplay /dev/hda1

-- Physical volume ---
PV Name                /dev/hda1
VG Name                myvg
PV Size                1.95 GB / NOT usable 4 MB [LVM: 122 KB]
PV#                    1
PV Status              available
Allocatable            yes (but full)
Cur LV                1
PE Size (KByte)        4096
Total PE               499
Free PE                0
Allocated PE           499
PV UUID                Sd44tK-9IRw-SrMC-M0kn-76iP-iftz-0VSen7
```

Se il volume fisico in questione è in uso, allora sarà necessario migrare i dati su di un altro volume fisico utilizzando il comando **pvmove**. Successivamente, utilizzate il comando **vgreduce** per rimuovere il volume fisico:

Il seguente comando rimuove il volume fisico **/dev/hda1** dal gruppo di volumi **my_volume_group**.

```
# vgreduce my_volume_group /dev/hda1
```

4.3.7. Modifica dei parametri di un gruppo di volumi

Il comando **vgchange** viene usato per disattivare ed attivare i gruppi di volumi come descritto in [Sezione 4.3.8, «Attivazione e disattivazione dei gruppi di volumi»](#). Usare questo comando per modificare alcuni dei parametri del gruppo per un gruppo esistente.

Il seguente comando modifica il numero massimo di logical volumes del gruppo di volumi **vg00** a 128.

```
vgchange -l 128 /dev/vg00
```

Per una descrizione dei parametri del gruppo di volumi modificabili con il comando **vgchange**, consultate la pagina man di **vgchange(8)**.

4.3.8. Attivazione e disattivazione dei gruppi di volumi

Quando create un gruppo di volumi esso per default viene attivato. Ciò significa che i volumi logici in quel gruppo sono accessibili e soggetti a modifiche.

In alcune situazioni sarà necessario disattivare un gruppo di volumi e renderli sconosciuti al kernel. Per disattivare o attivare un gruppo di volumi utilizzate l'argomento **-a (- -available)** del comando **vgchange**.

Il seguente esempio disattiva il gruppo di volumi **my_volume_group**.

```
vgchange -a n my_volume_group
```

Se il blocco clusterizzato è abilitato, aggiungete 'e' per attivare o disattivare un gruppo di volumi esclusivamente su di un nodo, oppure 'l' per attivare o disattivare un gruppo di volumi solo su di un nodo locale. I volumi logici con snapshot di un solo host, sono sempre attivati poichè possono essere usati su di un nodo per volta.

È possibile disattivare i volumi logici individuali con il comando `lvchange`, come descritto in [Sezione 4.4.8, «Modifica dei parametri di un gruppo di volumi logici»](#). Per informazioni su come attivare i volumi logici su nodi individuali in un cluster, consultate [Sezione 4.7, «Attivazione dei volumi logici su nodi individuali in un cluster»](#).

4.3.9. Rimozione dei gruppi di volumi

Per rimuovere un gruppo di volumi che non contiene alcun volume logico utilizzate il comando `vgremove`.

```
# vgremove officevg
Volume group "officevg" successfully removed
```

4.3.10. Separazione di un gruppo di volumi

Per separare i volumi fisici di un gruppo di volumi e crearne uno nuovo usate il comando `vgsplit`.

I volumi logici non possono essere divisi tra i gruppi di volumi. Ogni volume logico esistente deve essere presente interamente sui volumi fisici che formano il gruppo di volumi vecchio o quello nuovo. Se necessario tuttavia, è possibile utilizzare il comando `pvmove` per forzare la separazione.

Nel seguente esempio il nuovo gruppo di volumi `smallvg` viene separato da quello vecchio `bigvg`.

```
# vgsplit bigvg smallvg /dev/ram15
Volume group "smallvg" successfully split from "bigvg"
```

4.3.11. Come unire i gruppi di volumi

Per unire due gruppi di volumi in un gruppo di volumi singolo usare il comando `vgmerge`. Potrete unire un volume "sorgente" inattivo con un volume di "destinazione" attivo o inattivo, se le dimensioni dell'estensione fisica del volume risultano uguali e i sommari del volume logico e fisico di entrambi i gruppi di volumi sono adeguati ai limiti dei gruppi di volumi di destinazione.

Il seguente comando unisce il gruppo di volumi inattivo `my_vg`, nel gruppo di volumi attivo o inattivo `databases` fornendo informazioni verbose sul runtime.

```
vgmerge -v databases my_vg
```

4.3.12. Esecuzione del back up dei metadati del gruppo di volumi

I backup dei metadati e degli archivi sono eseguiti automaticamente ad ogni modifica della configurazione del gruppo di volumi e del volume logico, se non disabilitato, nel file `lvm.conf`. Per default, il backup dei metadati viene conservato in `/etc/lvm/backup` mentre gli archivi sono conservati su `/etc/lvm/archives`. È possibile eseguire il backup manuale dei metadati sul file `/etc/lvm/backup` attraverso il comando `vgcfgbackup`.

Il comando **vgcfrestore** ripristina i metadati di un gruppo di volumi dall'archivio su tutti i volumi fisici nei gruppi di volumi.

Per un esempio su come utilizzare il comando **vgcfgrestore** per recuperare i metadati del volume fisico, consultate la [Sezione 6.4, «Recupero dei metadati del volume fisico»](#).

4.3.13. Come rinominare un gruppo di volumi

Usate il comando **vgrename** per rinominare un gruppo di volumi esistente.

Entrambi i seguenti comandi rinominano il gruppo di volumi esistente **vg02** in **my_volume_group**

```
vgrename /dev/vg02 /dev/my_volume_group
```

```
vgrename vg02 my_volume_group
```

4.3.14. Come spostare un gruppo di volumi su di un altro sistema

È possibile spostare un intero gruppo di volumi LVM su di un altro sistema. Per fare questo è consigliato usare i comandi **vgexport** e **vgimport**.

vgexport rende un gruppo di volumi inattivo inaccessibile al sistema, ciò vi permetterà di distaccare i rispettivi volumi fisici. Il comando **vgimport** rende un gruppo di volumi accessibile ad una macchina dopo che il comando **vgexport** lo ha reso inattivo.

Per spostare un gruppo di volumi da un sistema ad un altro, eseguite le fasi di seguito riportate:

1. Assicuratevi che nessun utente stia cercando di accedere ai file sui volumi attivi nel gruppo di volumi, successivamente smontate i volumi logici.
2. Usate **-a n** del comando **vgchange** per segnare il gruppo di volumi come inattivo, questo impedisce qualsiasi attività sul gruppo in questione.
3. Usate il comando **vgexport** per esportare il gruppo di volumi. Ciò impedisce il suo accesso da parte del sistema dal quale lo state rimuovendo.

Dopo aver esportato il gruppo di volumi, il volume fisico verrà visualizzato in un gruppo di volumi esportato quando eseguirete il comando **pvscan**, come riportato nel seguente esempio.

```
[root@tng3-1]# pvscan
PV /dev/sda1    is in exported VG myvg [17.15 GB / 7.15 GB free]
PV /dev/sdc1    is in exported VG myvg [17.15 GB / 15.15 GB free]
PV /dev/sdd1    is in exported VG myvg [17.15 GB / 15.15 GB free]
...
```

Prima di arrestare il sistema sarà possibile scollegare i dischi che costituiscono il gruppo di volumi, ricollegandoli al nuovo sistema.

4. Una volta aver collegato i dischi al nuovo sistema, utilizzate il comando **vgimport** per importare il gruppo di volumi, rendendolo accessibile al nuovo sistema.
5. Attivate il gruppo di volumi con l'argomento **-a y** del comando **vgchange**.

6. Montate il file system in modo da renderlo accessibile ed utilizzabile.

4.3.15. Come ricreare una directory del gruppo di volumi

Per ricreare una directory del gruppo di volumi ed i file speciali del volume logico utilizzate **vgmknodes**. Questo comando controlla i file speciali di LVM2 nella directory **/dev** necessari per attivare i volumi logici. Esso crea qualsiasi file speciale mancante e rimuove quelli non utilizzati.

È possibile incorporare il comando **vgmknodes** nel comando **vgscan** specificando l'argomento **mknodes** sul comando **vgscan**.

4.4. AMMINISTRAZIONE DEL VOLUME LOGICO

Questa sezione descrive i comandi necessari per eseguire i compiti di amministrazione del volume logico.

4.4.1. Creazione di volumi logici lineari

Per creare un volume logico usare il comando **lvcreate**. Se non specificate un nome per il volume logico, verrà utilizzato il nome predefinito **lv1#**, dove **#** è il numero interno del volume logico.

Quando create un volume logico, esso viene ricavato da un gruppo di volumi utilizzando le estensioni disponibili sui volumi fisici che costituiscono il gruppo di volumi. Normalmente i volumi logici utilizzano tutto lo spazio disponibile sui lumi fisici relativi. Modificando il volume logico verrà liberato e riassegnato lo spazio nei volumi fisici.

Il seguente comando crea un volume logico di 10 gigabyte nel gruppo di volumi **vg1**.

```
lvcreate -L 10G vg1
```

Il seguente comando crea un volume logico lineare di 1500 megabyte chiamato **testlv** nel gruppo di volume **testvg**, creando il dispositivo a blocchi **/dev/testvg/testlv**.

```
lvcreate -L1500 -n testlv testvg
```

Il seguente comando crea un volume logico di 50 gigabyte chiamato **gfs1v** dalle tre estensioni nel gruppo di volumi **vg0**.

```
lvcreate -L 50G -n gfs1v vg0
```

Potrete utilizzare **-l** del comando **lvcreate** per specificare la dimensione in estensioni del volume logico. È possibile utilizzare questo argomento per specificare la percentuale del gruppo di volumi da utilizzare per il volume logico. Il seguente comando crea un volume logico chiamato **mylv** che utilizza il 60% dello spazio totale nel gruppo di volumi **testvol**.

```
lvcreate -l 60%VG -n mylv testvg
```

È possibile utilizzare anche **-l** del comando **lvcreate**, per specificare la percentuale dello spazio libero restante in un gruppo di volumi come dimensione del volume logico. Il seguente comando crea un volume logico chiamato **yourlv**, il quale utilizza tutto lo spazio non assegnato nel gruppo di volumi **testvol**.


```
lvcreate -l 100%FREE -n yourlv testvg
```

Usate **-l** del comando **lvcreate** per creare un volume logico che utilizzi l'intero gruppo di volumi. Un altro modo per creare un volume logico che utilizzi l'intero gruppo di volumi è tramite il comando **vgdisplay** per trovare la dimensione "PE totale", e utilizzare i risultati ottenuti come input per il comando **lvcreate**.

I seguenti comandi creano un volume logico chiamato **mylv** in grado di riempire il gruppo di volumi chiamato **testvg**.

```
# vgdisplay testvg | grep "Total PE"
Total PE                10230
# lvcreate -l 10230 testvg -n mylv
```

I volumi fisici usati per creare un volume logico possono essere molto importanti se il volume fisico deve essere rimosso, per questo motivo potreste considerare questa opportunità quando create il volume logico. Per informazioni su come rimuovere un volume fisico da un gruppo di volumi consultate la [Sezione 4.3.6, «Rimozione dei volumi fisici da un gruppo di volumi»](#).

Per creare un volume logico da assegnare da un volume fisico specifico al gruppo di volumi, specificate i volumi fisici alla fine della riga del comando **lvcreate**. Il seguente comando crea un logical volume chiamato **testlv** nel gruppo di volumi **testvg**, assegnato dal volume fisico **/dev/sdg1**,

```
lvcreate -L 1500 -ntestlv testvg /dev/sdg1
```

È possibile specificare quali estensioni di un volume fisico possono essere usate per un volume logico. Nel seguente esempio viene creato un volume logico lineare dalle estensioni 0 a 24 del volume fisico **/dev/sda1**, e dalle estensioni 50 fino a 124 del volume fisico **/dev/sdb1** nel gruppo di volumi **testvg**.

```
lvcreate -l 100 -n testlv testvg /dev/sda1:0-24 /dev/sdb1:50-124
```

Nel seguente esempio un volume logico lineare viene creato dalle estensioni 0 a 25 del volume fisico **/dev/sda1**, continuando la creazione del volume logico dall'estensione 100.

```
lvcreate -l 100 -n testlv testvg /dev/sda1:0-25:100-
```

La politica predefinita sull'assegnazione delle estensioni di un volume logico viene **ereditata**, per questo motivo viene applicata la stessa politica del gruppo di volumi. Queste politiche possono essere modificate utilizzando il comando **lvchange**. Per informazioni sulle policy di assegnazione consultate la [Sezione 4.3.1, «Creazione dei gruppi di volumi»](#).

4.4.2. Creazione dei volumi segmentati

Per processi molto grandi e sequenziali di lettura e scrittura, la creazione di un volume logico segmentato può migliorare l'efficienza dell'I/O dei dati. Per informazioni generali sui volumi segmentati, consultate la [Sezione 2.3.2, «Volumi logici segmentati»](#).

Durante la creazione di un volume logico segmentato, specificate il numero di segmenti con l'argomento **-i** del comando **lvcreate**. Ciò determina su quanti volumi fisici il volume logico verrà scritto. Il numero di segmenti non può essere maggiore del numero di volumi fisici presenti nel gruppo di volumi (se l'argomento **--alloc anywhere** non è stato utilizzato).

Se i dispositivi fisici che costituiscono un volume logico segmentato hanno dimensioni diverse, la dimensione massima del volume segmentato viene determinata dal dispositivo più piccolo. Per esempio in un segmento con due sezioni, la dimensione massima deve essere del doppio della dimensione del dispositivo più piccolo. In un segmento a tre sezioni, la dimensione massima è tre volte la dimensione del dispositivo più piccolo.

Il seguente comando crea un volume logico segmentato attraverso 2 volumi fisici con un segmento di 64KB. Il volume logico ha una dimensione di 50 gigabytes, è chiamato **gfs1v**, ed è ottenuto dal gruppo di volumi **vg0**.

```
lvcreate -L 50G -i2 -I64 -n gfs1v vg0
```

Come nei volumi lineari, è possibile specificare le estensioni del volume fisico utilizzato per il segmento. Il seguente comando crea un volume segmentato con una dimensione di 100 estensioni, è presente su due volumi fisici, chiamato **stripe1v**, e si trova nel gruppo di volumi **testvg**. Il segmento userà i settori 0-49 di **/dev/sda1** ed i settori 50-99 di **/dev/sdb1**.

```
# lvcreate -l 100 -i2 -n stripe1v testvg /dev/sda1:0-49 /dev/sdb1:50-99
Using default stripesize 64.00 KB
Logical volume "stripe1v" created
```

4.4.3. Creazione volumi speculari



NOTA

La creazione di un volume logico LVM speculare in un cluster ha bisogno di comandi e procedure simili usati nella creazione di un volume logico LVM speculare su di un nodo. Tuttavia per creare un volume LVM speculare in un cluster, il cluster e l'infrastruttura del cluster mirror devono essere in esecuzione, il cluster deve essere quorato ed il tipo di blocco nel file **lvm.conf** deve essere impostato correttamente in modo da abilitare il blocco del cluster. Per un esempio su come creare un volume speculare in un cluster consultare [Sezione 5.5, «Creazione di un volume logico LVM speculare in un cluster»](#).

Il tentativo di eseguire comandi multipli di conversione e creazione del mirror LVM in rapida successione da nodi multipli in un cluster potrebbe causare una congestione di questi comandi. Tale comportamento potrebbe causare la scadenza di alcune delle operazioni richieste e quindi il loro fallimento. Per evitare questo problema è consigliata l'esecuzione dei comandi di creazione del mirror del cluster da un solo nodo.

Durante la creazione di un volume speculare, specificate il numero di copie dei dati con l'opzione **-m** del comando **lvcreate**. Se usate **-m1** verrà creato un mirror, il quale dà luogo a due copie del file system: un volume logico lineare più una copia. Allo stesso modo, se specificate **-m2** verranno creati due mirror, dando luogo a tre copie del file system.

Il seguente comando crea un volume logico speculare con un mirror singolo. Il volume ha una dimensione di 50 gigabyte, chiamato **mirror1v**, e creato dal gruppo di volumi **vg0**:

```
lvcreate -L 50G -m1 -n mirror1v vg0
```

Un mirror LVM divide il dispositivo copiato in regioni le quali per impostazione predefinita hanno una dimensione di 512KB. Usare **-R** del comando **lvcreate** per specificare la dimensione della regione in MB. È possibile modificare la dimensione predefinita della regione modificando l'impostazione di **mirror_region_size** nel file **lvm.conf**.

NOTA

A causa di limitazioni presenti nella infrastruttura del cluster, i cluster mirror maggiori di 1.5 TB non potranno essere creati con una dimensione predefinita della regione pari a 512KB. Gli utenti che hanno necessità di avere mirror più grandi, dovranno aumentare la dimensione della regione predefinita ad una dimensione più grande. Il fallimento di tale operazione causerà una sospensione della creazione di LVM e dei rispettivi comandi.

Come linee guida generali per specificare una dimensione maggiore a 1.5TB prendete la dimensione in terabyte e arrotondate la cifra alla prossima potenza di due, usando quel numero come opzione **-R** del comando **lvcreate**. Per esempio, se la dimensione del mirror è 1.5TB specificare **-R 2**. Se la dimensione è di 3TB specificare **-R 4**. Per una dimensione pari a 5TB, specificare **-R 8**.

Il seguente comando crea un volume logico speculare con una regione con dimensione pari a 2MB:

```
lvcreate -m1 -L 2T -R 2 -n mirror vol_group
```

LVM mantiene un log molto piccolo usato per controllare quale regione è sincronizzata con i mirror. Per default il suddetto log viene conservato sul disco, mantenendolo uguale durante i vari processi di riavvio assicurando che il mirror non abbia bisogno di una nuova sincronizzazione ad ogni riavvio o crash della macchina. Se desiderate mantenere il suddetto log all'interno della memoria, allora potrete utilizzare l'opzione **--mirrorlog core**, ciò elimina la necessità di un dispositivo di log aggiuntivo, ma al tempo stesso avrà bisogno di una sincronizzazione dell'intero mirror ad ogni riavvio.

Il seguente comando crea un volume logico speculare dal gruppo di volumi **bigvg**. Il suddetto volume viene chiamato **ondiskmirvol** e possiede un mirror singolo. Il volume è di 12MB e mantiene il log del mirror all'interno della memoria.

```
# lvcreate -L 12MB -m1 --mirrorlog core -n ondiskmirvol bigvg
Logical volume "ondiskmirvol" created
```

Il mirror log viene creato su un dispositivo separato dai dispositivi sui quali viene creata qualsiasi sezione 'leg' del mirror. È possibile tuttavia, creare il mirror log sullo stesso dispositivo sul quale è stata creata la sezione del mirror, tramite l'argomento **--alloc anywhere** del comando **vgcreate**. Tale procedura potrebbe deteriorare le prestazioni, ma permette di creare un mirror anche se siete solo in possesso di due dispositivi.

Il seguente comando crea un volume logico speculare con un mirror singolo per il quale il log è sullo stesso dispositivo di uno dei segmenti del mirror. In questo esempio il gruppo di volumi **vg0** consiste di soli due dispositivi. Questo comando crea un volume di 500 megabyte chiamato **mirrorlv**, e creato dal gruppo di volumi **vg0**.

```
lvcreate -L 500M -m1 -n mirrorlv -alloc anywhere vg0
```



NOTA

Con mirror clusterizzati la gestione del log del mirror è di responsabilità del nodo del cluster con l'ID più basso. Per questo motivo quando il dispositivo con il log del mirror non è disponibile su di un sottoinsieme del cluster, il mirror clusterizzato può continuare ad operare senza alcun impatto fino a quando il nodo del cluster con l'ID più basso è in grado di accedere al log del mirror. Poichè il mirror non è distribuito nessuna azione automatica di correzione verrà emessa. Quando il nodo con l'ID più basso perde l'accesso al log del mirror allora l'azione automatica di correzione entrerà in azione (senza considerare l'accessibilità del log da altri nodi).

Per creare un log a sua volta speculare, specificare l'argomento `--mirrorlog mirrored`. Il seguente comando crea un volume logico speculare dal gruppo di volumi `bigvg`. Il suddetto volume viene chiamato `twologvol` e possiede un mirror singolo. Il volume è di 12MB ed il log del mirror è speculare, con ogni log archiviato su di un dispositivo separato.

```
# lvcreate -L 12MB -m1 --mirrorlog mirrored -n twologvol bigvg
Logical volume "twologvol" created
```

Proprio come con un log del mirror standard, è possibile creare log ridondanti sullo stesso dispositivo dei segmenti usando l'argomento `--alloc anywhere` del comando `vgcreate`. Tale procedura potrebbe deteriorare le prestazioni, ma permette di creare un log del mirror ridondante anche se non siete in possesso di un numero sufficiente di dispositivi per ogni log da archiviare su dispositivi separati rispetto ai segmenti del mirror.

Quando create un mirror le sue regioni vengono sincronizzate. Per componenti grandi del mirror il processo di sincronizzazione potrebbe richiedere un periodo lungo. Durante la creazione di un nuovo mirror che non necessita di essere attivato, potrete specificare l'opzione `nosync` in modo da indicare che la sincronizzazione iniziale dal primo dispositivo non risulta necessaria.

È possibile specificare il dispositivo da utilizzare per i log e per i segmenti del mirror insieme alle estensioni dei dispositivi. Per forzare il log su un particolare disco specificate in modo preciso l'estensione sul disco sul quale verrà ospitato. LVM non rispetterà necessariamente l'ordine attraverso il quale i dispositivi risultano elencati nella riga di comando. Se viene elencato un volume fisico, esso risulterà il solo luogo nel quale verrà eseguita l'assegnazione. Ogni estensione fisica inclusa nell'elenco precedentemente assegnata verrà ignorata.

Il seguente comando crea un volume logico speculare con un mirror singolo. La dimensione del volume è di 500 megabyte, il suo nome è `mirrorlv`, ed è stato ottenuto dal gruppo di volumi `vg0`. La prima sezione del mirror si trova sul dispositivo `/dev/sda1`, la seconda sul dispositivo `/dev/sdb1`, ed il log è su `/dev/sdc1`.

```
lvcreate -L 500M -m1 -n mirrorlv vg0 /dev/sda1 /dev/sdb1 /dev/sdc1
```

Il seguente comando crea un volume logico speculare con un singolo mirror. La dimensione del volume è di 500 megabyte, il suo nome è `mirrorlv`, ed è ottenuto dal gruppo di volumi `vg0`. La prima sezione del mirror è posizionata sulle estensioni da 0 a 499 del dispositivo `/dev/sda1`, la seconda sezione è posizionata sulle estensioni da 0 a 499 del dispositivo `/dev/sdb1`, ed il log del mirror inizia sull'estensione 0 di `/dev/sdc1`. Esse sono estensioni di 1MB. Se qualsiasi delle estensioni specificate è stata assegnata, essa verrà ignorata.

```
lvcreate -L 500M -m1 -n mirrorlv vg0 /dev/sda1:0-499 /dev/sdb1:0-499
/dev/sdc1:0
```



NOTA

Con Red Hat Enterprise Linux 6.1 è possibile combinare RAID0 (striping) e RAID1 (mirroring) in un unico volume logico. La creazione di un volume logico insieme alla determinazione del numero dei mirror (`--mirrors X`) ed al numero di segmenti (`--stripes Y`) genererà un dispositivo mirror con dispositivi segmentati.

4.4.3.1. Politica sugli errori del volume logico speculare

È possibile definire il comportamento di un volume logico speculare in presenza di un errore del dispositivo tramite i parametri `mirror_image_fault_policy` e `mirror_log_fault_policy` nella sezione `activation` del file `lvm.conf`. Dopo aver impostato i parametri su `remove`, il sistema cercherà di rimuovere il dispositivo che presenta l'errore e continuerà l'esecuzione senza di esso. Se il parametro è impostato su `allocate`, il sistema cercherà di rimuovere il dispositivo fallito ed assegnerà lo spazio su di un nuovo dispositivo come sostituto del dispositivo in questione; questa politica si comporterà come la politica `remove` se nessun dispositivo e spazio idonei possono essere assegnati come sostituti.

Per impostazione predefinita il parametro `mirror_log_fault_policy` è impostato su `allocate`. L'uso di questo tipo di politica per i log è veloce ed ha la possibilità di mantenere lo stato della sincronizzazione dopo ogni processo di riavvio o di crash. Se impostate questa politica su `remove`, quando un dispositivo fallisce il mirror utilizza un log presente in memoria ed il mirror non ricorderà il suo stato di sincronizzazione attraverso i processi di riavvio e di crash, eseguendo la risincronizzazione dell'intero mirror.

Per impostazione predefinita il parametro `mirror_image_fault_policy` è impostato su `remove`. Con questa politica se una immagine del mirror fallisce il mirror verrà convertito in un dispositivo non-speculare se è presente una copia idonea. La politica `allocate` per un dispositivo mirror risulta essere lenta poichè essa necessita di una risincronizzazione dei dispositivi da parte del mirror, al tempo stesso essa conserva la caratteristica del mirror del dispositivo.



NOTA

Al verificarsi di un fallimento del dispositivo con un mirror LVM verrà intrapreso un processo a due fasi di ripristino. Nella prima fase saranno rimossi i dispositivi falliti. Tale operazione potrebbe causare il passaggio del mirror ad un dispositivo lineare. Nella seconda fase se il parametro `mirror_log_fault_policy` è impostato su `allocate`, si cercherà di sostituire qualsiasi dispositivo fallito. Da notare tuttavia che non è possibile garantire durante la seconda fase l'implementazione di dispositivi precedentemente utilizzati dal mirror non compresi nel fallimento se altri dispositivi sono disponibili.

Per informazioni sul ripristino manuale da un errore LVM mirror consultare [Sezione 6.3, «Processo di recupero da un LVM Mirror Failure»](#).

4.4.3.2. Separazione di una immagine ridondante del volume logico speculare

È possibile separare una immagine ridondante di un volume logico speculare in modo da formare un nuovo volume logico. Per eseguire questa operazione usare l'opzione `--splitmirrors` del comando `lvconvert`, specificando il numero di immagini ridondanti da separare. Usare l'opzione `--name` del comando per specificare un nome per il volume logico appena separato.

Il seguente comando separa un nuovo volume logico **copy** dal volume logico speculare **vg/lv**. Il nuovo volume logico conterrà le 'gambe' o sezioni del mirror. In questo esempio LVM seleziona il dispositivo da separare.

```
lvconvert --splitmirrors 2 --name copy vg/lv
```

Sarà possibile specificare quale dispositivo separare utilizzando il seguente comando il quale è in grado di separare un nuovo volume logico chiamato **copy** dal volume logico speculare **vg/lv**. Il nuovo volume logico presenta due sezioni 'leg' del mirror le quali consistono nei dispositivi **/dev/sdc1** e **/dev/sde1**.

```
lvconvert --splitmirrors 2 --name copy vg/lv /dev/sd[ce]1
```

4.4.3.3. Correzione di un dispositivo logico speculare

Usare il comando **lvconvert --repair** per riparare un mirror dopo un errore del disco. Tale comando ritornerà il mirror in uno stato conforme. **lvconvert --repair** è un comando interattivo il quale richiederà all'utente di indicare se desidera sostituire i dispositivi falliti.

- Per saltare questa fase e sostituire tutti i dispositivi falliti specificare l'opzione **-y** sulla linea di comando.
- Per saltare questa fase e non sostituire alcun dispositivo fallito specificare l'opzione **-f** sulla linea di comando.
- Per saltare i prompt ed indicare politiche diverse per la sostituzione dell'immagine del mirror e del suo log specificare l'opzione **--use-policies** per usare le politiche di sostituzione del dispositivo specificate dai parametri **mirror_log_fault_policy** e **mirror_device_fault_policy** nel file **lvm.conf**.

4.4.3.4. Come modificare la configurazione del volume speculare

È possibile convertire un volume logico da un volume speculare ad uno lineare, o da un volume lineare ad uno speculare con il comando **lvconvert**. È possibile usare questo comando per riconfigurare altri parametri del mirror di un volume logico esistente, come ad esempio **corelog**.

Quando si esegue la conversione di un volume logico in un volume speculare, non si fa altro che creare delle sezioni o 'gambe' del mirror per un volume esistente. Ciò significa che il gruppo di volumi dovrà avere i dispositivi e lo spazio necessari per le sezioni del mirror e per il suo log.

Nel caso di perdita di un segmento del mirror, LVM converte il volume in un volume lineare in modo da poter ancora accedere al volume stesso senza il bisogno di alcuna ridondanza. Dopo la sostituzione della sezione è possibile utilizzare **lvconvert** per ripristinare il mirror. Questa procedura è presente in [Sezione 6.3, «Processo di recupero da un LVM Mirror Failure»](#).

Il seguente comando converte il volume logico lineare **vg00/lvol1** in un volume logico speculare.

```
lvconvert -m1 vg00/lvol1
```

Il seguente comando converte il volume logico speculare **vg00/lvol1** in un volume logico lineare rimuovendo la sezione del mirror.

```
lvconvert -m0 vg00/lvol1
```

4.4.4. Creazione dei volumi della snapshot

Usate `-s` del comando `lvcreate` per creare un volume della snapshot. Il suddetto volume è modificabile.



NOTA

Le snapshot LVM non sono supportate su tutti i nodi in un cluster. In un gruppo di volumi clusterizzati non sarà possibile creare un volume della snapshot. Con Red Hat Enterprise Linux 6.1 tuttavia se desiderate creare un backup di dati consistente su di un volume logico clusterizzato sarà possibile attivare il volume e successivamente creare la snapshot. Per informazioni su come attivare i volumi logici su un nodo consultate [Sezione 4.7, «Attivazione dei volumi logici su nodi individuali in un cluster»](#).



NOTA

Dalla release Red Hat Enterprise Linux 6.1 le snapshot LVM sono supportate per i volumi logici speculari.

Il seguente comando crea un volume logico della snapshot con una dimensione di 100 megabyte chiamato `/dev/vg00/snap`. Ciò creerà una snapshot del volume logico d'origine chiamata `/dev/vg00/lvol1`. Se il volume logico originale contiene un file system, allora sarà possibile montare il volume logico della snapshot su di una directory arbitraria, in modo da accedere ai contenuti del file system ed eseguire il backup durante l'aggiornamento del file system.

```
lvcreate --size 100M --snapshot --name snap /dev/vg00/lvol1
```

Dopo aver creato un volume logico della snapshot, `lvdisplay` dà luogo ad un output il quale include un elenco di tutti i logical volume della snapshot e del rispettivo stato (attivo o inattivo).

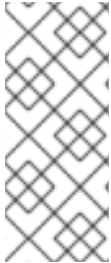
Il seguente esempio mostra uno stato del volume logico `/dev/new_vg/lvol0`, per il quale è stato creato un volume della snapshot `/dev/new_vg/newvgsnap`.

```
# lvdisplay /dev/new_vg/lvol0
--- Logical volume ---
LV Name                /dev/new_vg/lvol0
VG Name                new_vg
LV UUID                LBy1Tz-sr23-0jsI-LT03-nHLC-y8XW-EhC178
LV Write Access        read/write
LV snapshot status     source of
                        /dev/new_vg/newvgsnap1 [active]
LV Status              available
# open                 0
LV Size                52.00 MB
Current LE             13
Segments               1
Allocation             inherit
Read ahead sectors     0
Block device           253:2
```

Per default il comando `lvs` visualizza il volume d'origine e la percentuale corrente usata del volume della snapshot. Il seguente esempio mostra l'output predefinito per il comando `lvs` per un sistema che include il volume logico `/dev/new_vg/lvol0`, per il quale è stato creato un volume della snapshot

```
/dev/new_vg/newvgsnap.
```

```
# lvs
LV          VG      Attr   LSize   Origin Snap%   Move Log Copy%
lvol0       new_vg  owi-a- 52.00M
newvgsnap1  new_vg  swi-a-  8.00M lvol0    0.20
```



NOTA

Poichè la snapshot aumenta di misura al variare del volume d'origine, è importante controllare regolarmente la percentuale del volume della snapshot con il comando `lvs` in modo da assicurarsi che non venga del tutto utilizzata. Una snapshot utilizzata al 100% viene completamente persa poichè il processo di scrittura su sezioni non modificate dell'origine non andrà a buon fine senza corrompere la snapshot.

4.4.5. Unione dei volumi della snapshot

Con Red Hat Enterprise Linux 6 sarà possibile usare l'opzione `--merge` del comando `lvconvert` per eseguire il merge di una snapshot nel proprio volume d'origine. Se sia l'origine che il volume della snapshot non sono aperti il merge inizierà immediatamente. In caso contrario il merge inizierà quando la snapshot o l'origine saranno attivati per la prima volta e successivamente chiusi. Il merge di una snapshot in una origine che non può essere chiusa, per esempio un file system root, è rinviato fino alla successiva attivazione del volume d'origine. Nelle fasi iniziali del merge il volume logico risultante avrà il nome dell'origine, un numero minore ed un UUID. Durante il processo di merge i processi di lettura e scrittura sull'origine appariranno come se fossero diretti sulla snapshot unita. Al termine del processo la snapshot sulla quale è stato eseguito il merge verrà rimossa.

Il seguente comando unisce il volume della snapshot `vg00/lvol1_snap` con la propria origine.

```
lvconvert --merge vg00/lvol1_snap"
```

È possibile specificare snapshot multiple sulla linea di comando o utilizzare i tag dell'oggetto LVM per specificare l'unione di snapshot multiple con le rispettive origini. Nel seguente esempio i volumi logici `vg00/lvol1`, `vg00/lvol2`, e `vg00/lvol3` presenteranno tutti il seguente tag `@some_tag`". Il comando di seguito riportato eseguirà il merge dei volumi logici della snapshot per i tre volumi in modo sequenziale: `vg00/lvol1`, `vg00/lvol2`, e successivamente `vg00/lvol3`. Se è stata usata l'opzione `--background`, tutti i processi di merge del volume logico della snapshot inizieranno in parallelo.

```
lvconvert --merge @some_tag"
```

Per informazioni su come assegnare un tag agli oggetti LVM consultare [Appendice C, Tag oggetto LVM](#). Per maggiori informazioni sul comando `lvconvert --merge` consultare la pagina man di `lvconvert(8)`.

4.4.6. Numeri del dispositivo persistenti

I numeri maggiori e minori del dispositivo vengono assegnati dinamicamente durante il caricamento del modulo. Alcune applicazioni funzionano meglio se il dispositivo a blocchi è sempre attivato con lo stesso numero del dispositivo (maggiore e minore). Specificateli con `lvcreate` e `lvchange` utilizzando i seguenti argomenti:

```
--persistent y --major major --minor minor
```


Utilizzate un numero minore grande in modo da assicurarvi che non sia stato precedentemente assegnato dinamicamente ad un altro dispositivo.

Se state esportando un file system utilizzando NFS, specificando il parametro `fsid` nel file di esportazione, potreste non aver bisogno d'impostare un numero del dispositivo persistente all'interno di LVM.

4.4.7. Modifica della dimensione dei volumi logici

Per ridurre la dimensione di un volume logico utilizzate il comando `lvreduce`. Se il volume logico contiene un file system, assicuratevi di ridurre prima il file system (o utilizzate la GUI di LVM), in modo che il volume logico abbia una dimensione uguale a quella prevista dal file system.

Il seguente comando riduce la dimensione del volume logico `lv01` nel gruppo di volumi `vg00`, di 3 estensioni logiche.

```
lvreduce -l -3 vg00/lv01
```

4.4.8. Modifica dei parametri di un gruppo di volumi logici

Per modificare i parametri di un volume logico utilizzate il comando `lvchange`. Per un elenco di parametri modificabili consultate la man page di `lvchange(8)`.

Per attivare e disattivare i volumi logici potrete utilizzare il comando `lvchange`. Per attivare e disattivare tutti i volumi logici contemporaneamente in un gruppo di volumi, utilizzate `vgchange` come descritto in [Sezione 4.3.7, «Modifica dei parametri di un gruppo di volumi»](#).

Il seguente comando modifica i permessi sul volume `lv01` nel gruppo di volumi `vg00` in sola lettura.

```
lvchange -pr vg00/lv01
```

4.4.9. Modifica del nome dei volumi logici

Per modificare il nome di un volume logico esistente utilizzate il comando `lvrename`.

I seguenti comandi sono in grado di modificare il nome del volume logico `lvold` nel gruppo di volumi `vg02` in `lvnew`.

```
lvrename /dev/vg02/lvold /dev/vg02/lvnew
```

```
lvrename vg02 lvold lvnew
```

Per maggiori informazioni su come attivare i volumi logici sui nodi individuali in un cluster, consultate la [Sezione 4.7, «Attivazione dei volumi logici su nodi individuali in un cluster»](#).

4.4.10. Rimozione dei volumi logici

Per rimuovere un volume logico inattivo utilizzate il comando `lvremove`. Prima di poter eseguire la rimozione è necessario disattivare il volume logico. In aggiunta, in un ambiente clusterizzato è necessario disattivare un volume logico prima di poterlo rimuovere.

Il seguente comando rimuove il volume logico `/dev/testvg/testlv` dal gruppo di volumi `testvg`. Da notare che in questo caso il volume logico non è stato disattivato.

```
[root@tng3-1 lvm]# lvremove /dev/testvg/testlv
Do you really want to remove active logical volume "testlv"? [y/n]: y
Logical volume "testlv" successfully removed
```

È possibile disattivare esplicitamente il volume logico prima di rimuoverlo con il comando `lvchange -an`, in questo caso non sarete in grado di visualizzare il prompt attraverso il quale potete verificare se rimuovere un volume logico attivo.

4.4.11. Visualizzazione dei volumi logici

Per visualizzare le proprietà dei volumi logici LVM sono disponibili tre comandi: `lvs`, `lvdisplay`, e `lvscan`.

Il comando `lvs` fornisce le informazioni sul volume logico attraverso una forma configurabile mostrando una riga per volume logico. Il comando `lvs` è idoneo per un controllo del formato ed è utile per lo scripting. Per informazioni su come utilizzare il comando `lvs` per personalizzare l'output consultate la [Sezione 4.8, «Personalizzazione dei riporti per LVM»](#).

Il comando `lvdisplay` mostra le proprietà del volume logico (come ad esempio dimensione, schema e mappatura) in un formato fisso.

Il seguente comando mostra gli attributi di `lv012` in `vg00`. Se avete creato i volumi logici della snapshot per il volume logico originario, il comando mostrerà un elenco di tutti i volumi logici della snapshot e del loro stato (attivo o inattivo).

```
lvdisplay -v /dev/vg00/lv012
```

Il comando `lvscan` esegue la scansione di tutti i logical volume nel sistema elencondoli, in modo simile al seguente esempio.

```
# lvscan
ACTIVE                               '/dev/vg0/gfslv' [1.46 GB] inherit
```

4.4.12. Come aumentare la dimensione dei volumi logici

Per aumentare la dimensione di un volume logico utilizzate il comando `lvextend`.

Se desiderate estendere il volume logico sarà possibile indicare di quanto desiderate estenderlo, oppure specificare la grandezza dopo aver eseguito l'estensione.

Il seguente comando estende il volume logico `/dev/myvg/homevol` a 12 gigabyte.

```
# lvextend -L12G /dev/myvg/homevol
lvextend -- extending logical volume "/dev/myvg/homevol" to 12 GB
lvextend -- doing automatic backup of volume group "myvg"
lvextend -- logical volume "/dev/myvg/homevol" successfully extended
```

Il seguente comando aggiunge un altro gigabyte al volume logico `/dev/myvg/homevol`.

```
# lvextend -L+1G /dev/myvg/homevol
```

```
lvextend -- extending logical volume "/dev/myvg/homevol" to 13 GB
lvextend -- doing automatic backup of volume group "myvg"
lvextend -- logical volume "/dev/myvg/homevol" successfully extended
```

Come con il comando **lvcreate**, è possibile usare l'opzione **-l** del comando **lvextend** per specificare il numero di estensioni attraverso le quali aumentare la dimensione del volume logico. È possibile utilizzare questa opzione per specificare anche una percentuale del gruppo di volumi, o una percentuale di spazio libero disponibile nel gruppo di volumi. Il seguente comando estende il volume logico chiamato **testlv**, in modo da usare tutto lo spazio non assegnato nel gruppo di volumi **myvg**.

```
[root@tng3-1 ~]# lvextend -l +100%FREE /dev/myvg/testlv
Extending logical volume testlv to 68.59 GB
Logical volume testlv successfully resized
```

Dopo aver esteso il volume logico sarà necessario aumentare di conseguenza la dimensione del file system.

Per default, numerosi tool usati per la modifica della dimensione del file system aumentano la dimensione in modo da farla corrispondere alla dimensione del volume logico. In questo modo non vi sarà alcun bisogno di specificare la stessa dimensione in entrambi i comandi.

4.4.12.1. Come estendere un volume segmentato

Per aumentare la dimensione di un volume logico segmentato è necessario avere a disposizione spazio sufficiente sui volumi fisici che costituiscono il gruppo di volumi per supportare il segmento. Per esempio, se siete in possesso di un segmento a due vie la quale utilizza un intero gruppo di volumi, l'aggiunta di un volume fisico al gruppo di volumi non vi permetterà di estendere il segmento stesso. Al contrario, sarà necessario aggiungere almeno due volumi fisici al gruppo di volumi.

Per esempio, considerate un gruppo di volumi **vg** il quale consiste in due volumi fisici, come mostrato dal seguente comando **vgs**.

```
# vgs
VG      #PV #LV #SN Attr   VSize   VFree
vg       2   0   0 wz--n- 271.31G 271.31G
```

È possibile creare un segmento utilizzando l'intera quantità di spazio nel gruppo di volumi.

```
# lvcreate -n stripe1 -L 271.31G -i 2 vg
Using default stripesize 64.00 KB
Rounding up size to full physical extent 271.31 GB
Logical volume "stripe1" created
# lvs -a -o +devices
LV      VG      Attr   LSize   Origin Snap%  Move Log Copy%  Devices
stripe1 vg      -wi-a- 271.31G
/dev/sda1(0),/dev/sdb1(0)
```

Da notare che ora il gruppo di volumi non possiede più spazio libero.

```
# vgs
VG      #PV #LV #SN Attr   VSize   VFree
vg       2   1   0 wz--n- 271.31G    0
```

Il seguente comando aggiunge un altro volume fisico al gruppo di volumi, il quale avrà 135G di spazio aggiuntivo.

```
# vgextend vg /dev/sdc1
Volume group "vg" successfully extended
# vgs
VG    #PV #LV #SN Attr   VSize   VFree
vg      3  1  0 wz--n- 406.97G 135.66G
```

A questo punto non sarà possibile estendere il volume logico segmentato alla dimensione completa del gruppo di volumi, poichè i dispositivi interessati sono necessari per l'inserimento dei dati all'interno dei segmenti.

```
# lvextend vg/stripe1 -L 406G
Using stripesize of last segment 64.00 KB
Extending logical volume stripe1 to 406.00 GB
Insufficient suitable allocatable extents for logical volume stripe1:
34480
more required
```

Per estendere un volume logico segmentato, aggiungere prima un altro volume fisico e successivamente estendete il volume logico. In questo esempio, dopo l'aggiunta di due volumi fisici al gruppo di volumi, sarà possibile estendere il volume logico alla dimensione completa del gruppo di volumi.

```
# vgextend vg /dev/sdd1
Volume group "vg" successfully extended
# vgs
VG    #PV #LV #SN Attr   VSize   VFree
vg      4  1  0 wz--n- 542.62G 271.31G
# lvextend vg/stripe1 -L 542G
Using stripesize of last segment 64.00 KB
Extending logical volume stripe1 to 542.00 GB
Logical volume stripe1 successfully resized
```

Se non siete in possesso di un numero sufficiente di dispositivi fisici per poter estendere il volume logico segmentato, sarà comunque possibile estendere il volume se l'estensione in questione non è segmentata, tale procedura potrebbe dar luogo ad una prestazione non bilanciata. Durante l'aggiunta di spazio sul volume logico, l'operazione di default utilizza gli stessi parametri usati per lo strapping dell'ultimo segmento del volume logico esistente. Vi ricordiamo che i suddetti parametri possono essere sovrascritti. Nel seguente esempio il volume logico segmentato viene esteso in modo da utilizzare lo spazio libero restante dopo il fallimento del comando `lvextend` iniziale.

```
# lvextend vg/stripe1 -L 406G
Using stripesize of last segment 64.00 KB
Extending logical volume stripe1 to 406.00 GB
Insufficient suitable allocatable extents for logical volume stripe1:
34480
more required
# lvextend -i1 -l+100%FREE vg/stripe1
```

4.4.12.2. Estensione di un volume logico con politica di assegnazione `cling`

Durante l'estensione di un volume LVM sarà possibile usare l'opzione `--alloc cling` del comando

lvmextend per specificare la politica di assegnazione **cling**. Questa politica selezionerà lo spazio sugli stessi volumi fisici come segmento ultimo del volume logico esistente. Se lo spazio sui volumi fisici non è sufficiente e l'elenco dei tag è definito nel file **lvm.conf**, LVM controllerà la presenza di tag collegati ai volumi fisici e cercherà di corrispondere i tag dei volumi fisici tra estensioni esistenti e quelle nuove.

Per esempio, se siete in possesso di volumi logici speculari tra due siti all'interno di un gruppo singolo di volumi allora sarà possibile utilizzare i tag sui volumi fisici in base alla loro posizione con **@site1** e **@site2** specificando la seguente riga nel file **lvm.conf**:

```
cling_tag_list = [ "@site1", "@site2" ]
```

Per informazioni su come etichettare i volumi fisici consultare [Appendice C, Tag oggetto LVM](#).

Nel seguente esempio il file **lvm.conf** è stato modificato in modo da contenere la seguente riga:

```
cling_tag_list = [ "@A", "@B" ]
```

In questo esempio è stato creato un gruppo di volumi **taft** il quale è composto da volumi fisici **/dev/sdb1**, **/dev/sdc1**, **/dev/sdd1**, **/dev/sde1**, **/dev/sdf1**, **/dev/sdg1**, e **/dev/sdh1**. I suddetti volumi avranno i seguenti tag, **A**, **B**, e **C**. In questo esempio il tag **C** non viene utilizzato, ma in questo caso verrà mostrato che LVM utilizza i tag per selezionare i volumi fisici da usare per i segmenti 'o gambe' del mirror.

```
[root@taft-03 ~]# pvs -a -o +pv_tags /dev/sd[bcdefgh]1
PV          VG   Fmt  Attr PSize  PFree  PV Tags
/dev/sdb1   taft lvm2 a-   135.66g 135.66g A
/dev/sdc1   taft lvm2 a-   135.66g 135.66g B
/dev/sdd1   taft lvm2 a-   135.66g 135.66g B
/dev/sde1   taft lvm2 a-   135.66g 135.66g C
/dev/sdf1   taft lvm2 a-   135.66g 135.66g C
/dev/sdg1   taft lvm2 a-   135.66g 135.66g A
/dev/sdh1   taft lvm2 a-   135.66g 135.66g A
```

Il seguente comando crea un volume speculare di 100G dal gruppo di volumi **taft**.

```
[root@taft-03 ~]# lvcreate -m 1 -n mirror --nosync -L 100G taft
```

Il seguente comando mostra i dispositivi usati per le sezioni ed i log del mirror.

```
[root@taft-03 ~]# lvs -a -o +devices
LV          VG   Attr      LSize   Log           Copy%  Devices
mirror      taft Mwi-a-   100.00g mirror_mlog 100.00
mirror_mimage_0(0),mirror_mimage_1(0)
[mirror_mimage_0] taft      iwi-ao 100.00g
/dev/sdb1(0)
[mirror_mimage_1] taft      iwi-ao 100.00g
/dev/sdc1(0)
[mirror_mlog]    taft      lwi-ao   4.00m
/dev/sdh1(0)
```

Il seguente comando estende la dimensione del volume speculare utilizzando la politica di assegnazione **cling** per indicare che i segmenti del mirror devono essere estesi utilizzando i volumi fisici con lo stesso tag.

```
[root@taft-03 ~]# lvextend --alloc cling -L +100G taft/mirror
Extending 2 mirror images.
Extending logical volume mirror to 200.00 GiB
Logical volume mirror successfully resized
```

Il seguente comando mostra l'estensione dei segmenti del mirror utilizzando i volumi fisici con lo stesso tag del segmento. Da notare come i volumi fisici con lo stesso tag, in questo caso **C**, sono stati ignorati.

```
[root@taft-03 ~]# lvs -a -o +devices
  LV          VG      Attr   LSize   Log           Copy%  Devices
  mirror      taft    Mwi-a- 200.00g mirror_mlog   50.16
mirror_mimage_0(0),mirror_mimage_1(0)
  [mirror_mimage_0] taft    Iwi-ao 200.00g
/dev/sdb1(0)
  [mirror_mimage_0] taft    Iwi-ao 200.00g
/dev/sdg1(0)
  [mirror_mimage_1] taft    Iwi-ao 200.00g
/dev/sdc1(0)
  [mirror_mimage_1] taft    Iwi-ao 200.00g
/dev/sdd1(0)
  [mirror_mlog]   taft    lwi-ao  4.00m
/dev/sdh1(0)
```

4.4.13. Come ridurre la dimensione dei volumi logici

Per ridurre la dimensione di un volume logico smontate prima il file system. Successivamente utilizzate il comando **lvreduce** per ridurre il volume. Dopo aver ridotto il volume, rimontate il file system.



AVVERTIMENTO

È importante ridurre la dimensione del file system o di qualsiasi entità presente nel volume prima di diminuire il volume stesso, in caso contrario rischierete di perdere i dati.

La diminuzione del volume logico libererà parte del gruppo di volumi in modo da poter riassegnarlo ad altri volumi logici nel gruppo stesso.

Nel seguente esempio la dimensione del volume logico **lv011** nel gruppo di volumi **vg00** viene ridotta di 3 estensioni logiche.

```
lvreduce -l -3 vg00/lv011
```

4.5. CONTROLLO DELLE SCANSIONI DEL DISPOSITIVO LVM CON I FILTRI

All'avvio il comando **vgscan** esegue la scansione dei dispositivi a blocchi sul sistema andando alla

ricerca di etichette LVM per determinare i volumi fisici, leggere i metadati, e creare un elenco di gruppi di volumi. I nomi dei volumi fisici sono archiviati nel file di cache di ogni nodo presente nel sistema, `/etc/lvm/.cache`. I comandi che ne seguono possono leggere il file in modo da evitarne successive scansioni.

Potrete controllare quale dispositivo LVM è in grado di eseguire la scansione attraverso l'impostazione dei filtri nel file di configurazione `lvm.conf`. I filtri in `lvm.conf` consistono in una serie di espressioni regolari semplici, applicate ai nomi del dispositivo nella directory `/dev`, per decidere se accettare o rifiutare ogni dispositivo a blocchi trovato.

I seguenti esempi mostrano l'utilizzo dei filtri per controllare su quali dispositivi viene eseguita la scansione da parte di LVM. Da notare che alcuni di questi esempi non rappresentano necessariamente l'approccio migliore, poichè le espressioni regolari corrispondono al nome del percorso. Per esempio `a/loop/` è equivalente a `a/. *loop. */` e corrisponderebbe a `/dev/solooperation/lvol1`.

Il seguente filtro aggiunge tutti i dispositivi trovati, e rappresenta il comportamento predefinito poichè non vi è altro filtro configurato nel file di configurazione:

```
filter = [ "a/. */" ]
```

Il seguente filtro rimuove il dispositivo `cdrom` in modo da evitare rallentamenti se il drive non contiene alcun media:

```
filter = [ "r|/dev/cdrom|" ]
```

Il seguente filtro aggiunge tutti i loop e rimuove tutti gli altri dispositivi a blocchi:

```
filter = [ "a/loop.*/", "r/. */" ]
```

Il seguente filtro aggiunge tutti i loop e IDE, rimuovendo tutti gli altri dispositivi a blocchi:

```
filter =[ "a|loop.*|", "a|/dev/hd.*|", "r|. */" ]
```

Il seguente filtro aggiunge solo la partizione 8 sul primo drive IDE, e rimuove tutti gli altri dispositivi a blocchi:

```
filter = [ "a|^/dev/hda8$|", "r/. */" ]
```

Per maggiori informazioni sul file `lvm.conf`, consultate la pagina man di `lvm.conf(5)` e [Appendice B, File di configurazione LVM](#)

4.6. RIPOSIZIONAMENTO DATI ONLINE

È possibile spostare i dati online durante l'utilizzo del vostro sistema utilizzando il comando `pvmove`.

Il comando `pvmove` divide i dati da spostare in sezioni, e crea un mirror provvisorio per poter muovere ogni sezione. Per maggiori informazioni sul funzionamento di `pvmove`, consultate la pagina man di `pvmove(8)`.

Il seguente comando sposta tutto lo spazio assegnato dal volume fisico `/dev/sdc1` ad altri volumi fisici presenti nel gruppo di volumi:

```
pvmove /dev/sdc1
```

Il seguente comando sposta solo le estensioni del volume logico **MyLV**.

```
pvmove -n MyLV /dev/sdc1
```

Poichè l'esecuzione del comando **pvmove** potrebbe richiedere molto tempo, è consigliato eseguirlo nel background in modo da evitare la visualizzazione del progresso degli aggiornamenti. Il seguente comando sposta tutte le estensioni assegnate al volume fisico **/dev/sdc1** su **/dev/sdf1** nel background.

```
pvmove -b /dev/sdc1 /dev/sdf1
```

Il seguente comando riporta il progresso relativo al processo di spostamento per mezzo di un valore percentuale, ad intervalli di cinque secondi.

```
pvmove -i5 /dev/sdd1
```

4.7. ATTIVAZIONE DEI VOLUMI LOGICI SU NODI INDIVIDUALI IN UN CLUSTER

Se avete installato LVM in un ambiente cluster, molto probabilmente avrete bisogno di attivare i volumi logici esclusivamente su di un nodo.

Per attivare i volumi logici solo su di un nodo utilizzate il comando **lvchange -aey**. Alternativamente, potrete utilizzare **lvchange -aly** per attivare i volumi logici solo, ma non esclusivamente, sul nodo locale. Successivamente potrete attivarli simultaneamente su nodi aggiuntivi.

È possibile attivare i volumi logici su singoli nodi utilizzando i tag LVM descritti in [Appendice C, Tag oggetto LVM](#). È possibile specificare altresì l'attivazione dei nodi nel file di configurazione, come descritto in [Appendice B, File di configurazione LVM](#).

4.8. PERSONALIZZAZIONE DEI RIPORTI PER LVM

È possibile produrre riporti personalizzabili e brevi sugli oggetti LVM attraverso i comandi **pvs**, **lvs**, e **vgs**. I riporti generati da questi comandi includono un output di una sola riga per ogni oggetto. Ogni riga contiene un elenco ordinato di campi relativi alle proprietà dell'oggetto. È possibile selezionare gli oggetti da riportare in cinque modi diversi: attraverso il volume fisico, il gruppo di volumi, volume logico, il segmento del volume fisico ed il segmento del volume logico.

Le seguenti sezioni forniscono:

- Un sommario degli argomenti del comando utilizzabili per controllare il formato del riporto generato.
- Un elenco dei campi selezionabili per ogni oggetto LVM.
- Un sommario degli argomenti del comando utilizzabili per ordinare il riporto generato.
- Le informazioni su come specificare le unità dell'output del riporto.

4.8.1. Controllo del formato

Se utilizzate il comando **pvs**, **lvs**, o **vgs** determinerete il set predefinito dei campi visualizzati ed il tipo di ordine da seguire. Potrete controllare l'output dei suddetti comandi con i seguenti argomenti:

- È possibile modificare i campi visualizzati in modo da avere una selezione diversa dall'impostazione predefinita utilizzando l'opzione **-o**. Per esempio, il seguente output è il display predefinito per il comando **pvs** (il quale mostra le informazioni sui volumi fisici).

```
# pvs
PV          VG      Fmt  Attr PSize  PFree
/dev/sdb1   new_vg  lvm2 a-   17.14G 17.14G
/dev/sdc1   new_vg  lvm2 a-   17.14G 17.09G
/dev/sdd1   new_vg  lvm2 a-   17.14G 17.14G
```

È possibile visualizzare solo la dimensione ed il nome del volume fisico con il seguente comando.

```
# pvs -o pv_name,pv_size
PV          PSize
/dev/sdb1   17.14G
/dev/sdc1   17.14G
/dev/sdd1   17.14G
```

- È possibile aggiungere un campo all'output utilizzando il segno più (+); il suddetto segno viene usato in combinazione con l'opzione **-o**.

Il seguente esempio mostra l'UUID del physical volume insieme ai campi predefiniti.

```
# pvs -o +pv_uuid
PV          VG      Fmt  Attr PSize  PFree  PV UUID
/dev/sdb1   new_vg  lvm2 a-   17.14G 17.14G onFF2w-1fLC-ughJ-D9eB-
M7iv-6XqA-dqGeXY
/dev/sdc1   new_vg  lvm2 a-   17.14G 17.09G Joqlch-yWSj-kuEn-IdwM-
01S9-X08M-mcpsVe
/dev/sdd1   new_vg  lvm2 a-   17.14G 17.14G yvfvZK-Cf31-j75k-dECm-
0RZ3-0dGW-UqkCS
```

- Aggiungendo l'opzione **-v** ad un comando, includerete alcuni campi aggiuntivi. Per esempio, il comando **pvs -v** mostrerà i campi **DevSize** e **PV UUID** insieme ai campi predefiniti.

```
# pvs -v
Scanning for physical volume names
PV          VG      Fmt  Attr PSize  PFree  DevSize PV UUID
/dev/sdb1   new_vg  lvm2 a-   17.14G 17.14G  17.14G onFF2w-1fLC-
ughJ-D9eB-M7iv-6XqA-dqGeXY
/dev/sdc1   new_vg  lvm2 a-   17.14G 17.09G  17.14G Joqlch-yWSj-
kuEn-IdwM-01S9-X08M-mcpsVe
/dev/sdd1   new_vg  lvm2 a-   17.14G 17.14G  17.14G yvfvZK-Cf31-
j75k-dECm-0RZ3-0dGW-tUqkCS
```

- L'opzione **--noheadings** rimuove la riga delle intestazioni. Ciò può essere utile per la scrittura di script.

Il seguente esempio utilizza l'opzione **--noheadings** insieme a **pv_name**, generando un elenco di tutti i physical volume.

```
# pvs --noheadings -o pv_name
/dev/sdb1
/dev/sdc1
/dev/sdd1
```

- L'opzione **--separator *separator*** utilizza *separator* per separare ogni campo.

Il seguente esempio separa i campi predefiniti per l'output di **pvs** con il segno uguale (=).

```
# pvs --separator =
PV=VG=Fmt=Attr=PSize=PFree
/dev/sdb1=new_vg=lvm2=a-=17.14G=17.14G
/dev/sdc1=new_vg=lvm2=a-=17.14G=17.09G
/dev/sdd1=new_vg=lvm2=a-=17.14G=17.14G
```

Per mantenere i campi allineati durante l'utilizzo del **separator**, utilizzate l'opzione **separator** insieme con **--aligned**.

```
# pvs --separator = --aligned
PV          =VG      =Fmt =Attr=PSize =PFree
/dev/sdb1 =new_vg=lvm2=a- =17.14G=17.14G
/dev/sdc1 =new_vg=lvm2=a- =17.14G=17.09G
/dev/sdd1 =new_vg=lvm2=a- =17.14G=17.14G
```

Potrete usare **-P** del comando **lvs** o **vgs**, per mostrare le informazioni relative ad un volume fallito che potrebbero non comparire nell'output. Per informazioni sull'output generato da questa opzione consultate [Sezione 6.2, «Come visualizzare le informazioni su dispositivi falliti»](#).

Per un elenco completo delle opzioni consultate le pagine man di **pvs(8)**, **vgs(8)** e **lvs(8)**.

I campi del gruppo di volumi possono essere uniti con i campi del volume fisico (e segmento del volume fisico), o con i campi del volume logico (e segmento del volume logico), al contrario i campi del volume fisico e del volume logico non possono essere uniti tra loro. Per esempio, il seguente comando mostrerà una riga per ogni volume fisico.

```
# vgs -o +pv_name
VG      #PV #LV #SN Attr   VSize  VFree  PV
new_vg   3   1   0 wz--n- 51.42G 51.37G /dev/sdc1
new_vg   3   1   0 wz--n- 51.42G 51.37G /dev/sdd1
new_vg   3   1   0 wz--n- 51.42G 51.37G /dev/sdb1
```

4.8.2. Selezione dell'oggetto

Questa sezione fornisce una serie di tabelle contenenti le informazioni visualizzabili relative agli oggetti LVM, attraverso l'utilizzo dei comandi **pvs**, **vgs**, e **lvs**.

Per convenienza, il prefisso del nome del campo può essere escluso se corrisponde al default del comando in questione. Per esempio, con il comando **pvs**, **name** significa **pv_name**, mentre con il comando **vgs**, **name** viene interpretato come **vg_name**.

L'esecuzione del seguente comando risulta essere equivalente all'esecuzione del comando **pvs -o pv_free**.

```
# pvs -o +free
PFree
17.14G
17.09G
17.14G
```

Il comando pvs

Tabella 4.1, «campi relativi a pvs» elenca le opzioni relative al comando `pvs`, insieme al nome del campo visualizzato nel display d'intestazione, ed una descrizione del campo.

Tabella 4.1. campi relativi a pvs

Opzione	Intestazione	Descrizione
<code>dev_size</code>	DevSize	Misura del dispositivo sul quale è stato creato il volume fisico
<code>pe_start</code>	1st PE	L'offset all'inizio della prima estensione fisica del dispositivo interessato
<code>pv_attr</code>	Attr	Stato del volume fisico: (a)ssegnabile o e(s)portato
<code>pv_fmt</code>	Fmt	Il formato dei metadati del volume fisico (lvm2 o lvm1)
<code>pv_free</code>	PFree	Lo spazio libero restante sul volume fisico
<code>pv_name</code>	PV	Il nome del volume fisico
<code>pv_pe_alloc_count</code>	Alloc	Numero di estensioni fisiche usate
<code>pv_pe_count</code>	PE	Numero di estensioni fisiche
<code>pvseg_size</code>	SSize	La dimensione del segmento del volume fisico
<code>pvseg_start</code>	Start	L'estensione fisica iniziale del segmento del volume fisico
<code>pv_size</code>	PSize	La dimensione del volume fisico
<code>pv_tags</code>	PV Tags	Le etichette di LVM relative al volume fisico
<code>pv_used</code>	Used	La quantità di spazio attualmente usato sul volume fisico
<code>pv_uuid</code>	PV UUID	L'UUID del volume fisico

Il comando `pvs` mostra per default i seguenti campi: `pv_name`, `vg_name`, `pv_fmt`, `pv_attr`, `pv_size`, `pv_free`. L'ordine seguito viene indicato dall'opzione `pv_name`.

```
# pvs
PV          VG      Fmt  Attr PSize  PFree
/dev/sdb1   new_vg  lvm2 a-   17.14G 17.14G
/dev/sdc1   new_vg  lvm2 a-   17.14G 17.09G
/dev/sdd1   new_vg  lvm2 a-   17.14G 17.13G
```

Utilizzando **-v** con il comando **pvs**, aggiungerete i seguenti campi al display predefinito: **dev_size**, **pv_uuid**.

```
# pvs -v
    Scanning for physical volume names
PV          VG      Fmt  Attr PSize  PFree  DevSize PV UUID
/dev/sdb1   new_vg  lvm2 a-   17.14G 17.14G  17.14G onFF2w-1fLC-ughJ-D9eB-
M7iv-6XqA-dqGeXY
/dev/sdc1   new_vg  lvm2 a-   17.14G 17.09G  17.14G Joqlch-yWSj-kuEn-IdwM-
01S9-X08M-mcpsVe
/dev/sdd1   new_vg  lvm2 a-   17.14G 17.13G  17.14G yvfvZK-Cf31-j75k-dECm-
0RZ3-0dGW-tUqKCS
```

Usate **--segments** del comando **pvs**, per visualizzare le informazioni su ogni segmento del volume fisico. Un segmento è rappresentato da un gruppo di estensioni. Una vista del segmento può essere utile per controllare se il vostro volume logico è frammentato.

Il comando **pvs --segments** mostra per default i seguenti campi: **pv_name**, **vg_name**, **pv_fmt**, **pv_attr**, **pv_size**, **pv_free**, **pvseg_start**, **pvseg_size**. L'elenco è ordinato in base al **pv_name** e **pvseg_size** all'interno del physical volume.

```
# pvs --segments
PV          VG          Fmt  Attr PSize  PFree  Start SSize
/dev/hda2   VolGroup00  lvm2 a-   37.16G 32.00M    0  1172
/dev/hda2   VolGroup00  lvm2 a-   37.16G 32.00M  1172   16
/dev/hda2   VolGroup00  lvm2 a-   37.16G 32.00M  1188    1
/dev/sda1   vg          lvm2 a-   17.14G 16.75G    0   26
/dev/sda1   vg          lvm2 a-   17.14G 16.75G   26   24
/dev/sda1   vg          lvm2 a-   17.14G 16.75G   50   26
/dev/sda1   vg          lvm2 a-   17.14G 16.75G   76   24
/dev/sda1   vg          lvm2 a-   17.14G 16.75G  100   26
/dev/sda1   vg          lvm2 a-   17.14G 16.75G  126   24
/dev/sda1   vg          lvm2 a-   17.14G 16.75G  150   22
/dev/sda1   vg          lvm2 a-   17.14G 16.75G  172  4217
/dev/sdb1   vg          lvm2 a-   17.14G 17.14G    0  4389
/dev/sdc1   vg          lvm2 a-   17.14G 17.14G    0  4389
/dev/sdd1   vg          lvm2 a-   17.14G 17.14G    0  4389
/dev/sde1   vg          lvm2 a-   17.14G 17.14G    0  4389
/dev/sdf1   vg          lvm2 a-   17.14G 17.14G    0  4389
/dev/sdg1   vg          lvm2 a-   17.14G 17.14G    0  4389
```

Per visualizzare i dispositivi rilevati da LVM e non ancora inizializzati come volumi fisici LVM, usate il comando **pvs -a**.

```
# pvs -a
PV          VG          Fmt  Attr PSize  PFree
/dev/VolGroup00/LogVol01  --      0      0
/dev/new_vg/lvol0         --      0      0
```

```

/dev/ram          --      0      0
/dev/ram0         --      0      0
/dev/ram2         --      0      0
/dev/ram3         --      0      0
/dev/ram4         --      0      0
/dev/ram5         --      0      0
/dev/ram6         --      0      0
/dev/root         --      0      0
/dev/sda          --      0      0
/dev/sdb          --      0      0
/dev/sdb1         new_vg  lvm2  a-   17.14G 17.14G
/dev/sdc          --      0      0
/dev/sdc1         new_vg  lvm2  a-   17.14G 17.09G
/dev/sdd          --      0      0
/dev/sdd1         new_vg  lvm2  a-   17.14G 17.14G

```

Il comando vgs

Tabella 4.2. «Campi relativi a vgs» elenca le opzioni relative al comando **vgs**, insieme al nome del campo visualizzato nel display d'intestazione, ed una descrizione del campo.

Tabella 4.2. Campi relativi a vgs

Opzione	Intestazione	Descrizione
lv_count	#LV	Il numero di volumi logici contenuti nel gruppo di volumi
max_lv	MaxLV	Il numero massimo di volumi logici permessi nel gruppo di volumi (0 se illimitato)
max_pv	MaxPV	Il numero massimo di physical volume permessi nel gruppo di volumi (0 se illimitato)
pv_count	#PV	Il numero di physical volume che definisce il gruppo di volumi
snap_count	#SN	Il numero di snapshot contenute nel gruppo di volumi
vg_attr	Attr	Stato del gruppo di volumi: (w)riteable, (r)eadonly, (r)esi(z)eable, (e(x)ported, (p)artial e (c)lustered.
vg_extent_count	#Ext	Il numero di estensioni fisiche nel gruppo di volumi
vg_extent_size	Ext	La dimensione delle estensioni fisiche nel gruppo di volumi
vg_fmt	Fmt	Il formato dei metadati del gruppo di volumi (lvm2 or lvm1)
vg_free	VFree	La dimensione dello spazio libero restante nel gruppo di volumi

Opzione	Intestazione	Descrizione
vg_free_count	Free	Il numero di estensioni fisiche libere presenti nel gruppo di volumi
vg_name	VG	Il nome del gruppo di volumi
vg_seqno	Seq	Il numero che rappresenta la revisione del gruppo di volumi
vg_size	VSize	La dimensione del gruppo di volumi
vg_sysid	SYS ID	ID del sistema LVM1
vg_tags	VG Tags	Le etichette di LVM relative al gruppo di volumi
vg_uuid	VG UUID	L'UUID del gruppo di volumi

Il comando **vgs** visualizza per default i seguenti campi: **vg_name**, **pv_count**, **lv_count**, **snap_count**, **vg_attr**, **vg_size**, **vg_free**. L'elenco viene ordinato in base al **vg_name**.

```
# vgs
VG      #PV #LV #SN Attr   VSize  VFree
new_vg   3   1   1 wz--n- 51.42G 51.36G
```

Usando **-v** del comando **vgs**, verranno aggiunti all'impostazione predefinita i seguenti campi: **vg_extent_size**, **vg_uuid**.

```
# vgs -v
Finding all volume groups
Finding volume group "new_vg"
VG      Attr   Ext   #PV #LV #SN VSize  VFree  VG UUID
new_vg wz--n- 4.00M   3   1   1 51.42G 51.36G jxQJ0a-ZKk0-OpMO-0118-
nlw0-wwqd-fD5D32
```

Il comando **lvs**

[Tabella 4.3, «Campi relativi a lvs»](#) elenca le opzioni relative al comando **lvs**, insieme al nome del campo visualizzato nel display d'intestazione, ed una descrizione del campo.

Tabella 4.3. Campi relativi a lvs

Opzione	Intestazione	Descrizione
chunksize	Chunk	Dimensione unità in un volume della snapshot
chunk_size		

Opzione	Intestazione	Descrizione
copy_percent	Copy%	La percentuale di sincronizzazione di un volume logico speculare; usato anche quando le estensioni fisiche vengono spostate con il comando pv_move
devices	Dispositivi	I dispositivi che costituiscono il volume logico: i volumi fisici, volumi logici, e le estensioni fisiche e logiche iniziali
lv_attr	Attr	<p>Stato del volume logico. Di seguito vengono riportati i bit per l'attributo del volume logico:</p> <div> <p>Bit 1: Tipo di volume: (m)speculare, (M)speculare senza sincronizzazione iniziale, (o)rigine, (p)vmove, (s)snapshot, (S)snapshot invalida, (v)irtuale</p> <p>Bit2: Permessi: (w)scrivibile, (r)solo-lettura</p> <p>Bit 3: Policy per l'assegnazione: (c)ontiguous, (n)ormal, (a)nywhere, (i)nherited. Potrete trarne vantaggio se il volume è stato bloccato nei confronti delle modifiche relative all'assegnazione, per esempio durante l'esecuzione del comando pvmove.</p> <p>Bit 4: (m)inore fisso</p> <p>Bit 5 Stato: (a)ttivo, (s)ospeso, (l)snapshot invalida, (S)snapshot invalida sospesa, (d)dispositivo mappato presente senza tabelle, dispositivo mappato presente con tabella (i)nattiva</p> <p>Bit 6: dispositivo (o)aperto</p> </div>
lv_kernel_major	KMaj	Numero maggiore del dispositivo del volume logico (-1 se inattivo)
lv_kernel_minor	KMIN	Numero minore del dispositivo del volume logico (-1 se inattivo)
lv_major	Maj	Il numero maggiore persistente del dispositivo del volume logico (-1 se non specificato)
lv_minor	Min	Il numero minore persistente del dispositivo del volume logico (-1 se non specificato)
lv_name	LV	Il nome del volume logico
lv_size	LSize	La dimensione del volume logico

Opzione	Intestazione	Descrizione
lv_tags	LV Tags	Le etichette di LVM relative al volume logico
lv_uuid	LV UUID	L'UUID del volume logico.
mirror_log	Log	Dispositivo nel quale risiede il log del mirror
modules	Moduli	Il target device-mapper del kernel corrispondente necessario per utilizzare questo volume logico
move_pv	Move	Volume fisico sorgente di un volume logico provvisorio creato con il comando pvmove
origin	Origine	Il dispositivo d'origine di un volume della snapshot
<div>regionsize</div> <div>region_size</div>	Regione	La dimensione dell'unità di un volume logico speculare
seg_count	#Seg	Il numero di segmenti nel volume logico
seg_size	SSize	La dimensione dei segmenti nel volume logico
seg_start	Start	L'offset del segmento nel volume logico
seg_tags	Seg Tags	Le etichette di LVM relative ai segmenti del volume logico
segtype	Tipo	IL tipo di segmento di un volume logico (per esempio: mirror, segmentato, lineare)
snap_percent	Snap%	Percentuale corrente del volume della snapshot in uso
stripes	#Str	Numero di segmenti o mirror in un volume logico
<div>stripesize</div> <div>stripe_size</div>	Segmento	Dimensione dell'unità del segmento in un volume logico segmentato

Il comando **lvs** mostra per default i seguenti campi: **lv_name**, **vg_name**, **lv_attr**, **lv_size**, **origin**, **snap_percent**, **move_pv**, **mirror_log**, **copy_percent**. L'elenco viene visualizzato in base al **vg_name** e **lv_name** all'interno del gruppo di volumi.

```
# lvs
LV          VG      Attr      LSize   Origin Snap%   Move Log Copy%
```



```
lv100      new_vg owi-a- 52.00M
newvgsnap1 new_vg swi-a- 8.00M lv100      0.20
```

Utilizzando **-v** con il comando **lvs** aggiungerete i seguenti campi all'impostazione predefinita: **seg_count**, **lv_major**, **lv_minor**, **lv_kernel_major**, **lv_kernel_minor**, **lv_uuid**.

```
# lvs -v
    Finding all logical volumes
  LV          VG      #Seg Attr   LSize  Maj Min KMaj KMin Origin Snap%
Move Copy%   Log LV UUID
lv100        new_vg    1 owi-a- 52.00M  -1  -1 253  3
LBy1Tz-sr23-0jsI-LT03-nHLC-y8XW-EhC178
newvgsnap1   new_vg    1 swi-a- 8.00M  -1  -1 253  5   lv100    0.20
1ye10U-1cIu-o79k-20h2-ZGF0-qCJm-CfbsIx
```

Per visualizzare le informazioni contenute nelle colonne predefinite le quali contengono a loro volta le informazioni sul segmento, utilizzate **--segments** del comando **lvs**. Quando utilizzate l'opzione **segments** il prefisso **seg** diventa facoltativo. Il comando **lvs --segments** mostra per default i seguenti campi: **lv_name**, **vg_name**, **lv_attr**, **stripes**, **segtype**, **seg_size**. Il display predefinito viene visualizzato in base al **vg_name**, **lv_name** all'interno del gruppo di volumi, e **seg_start** all'interno del volume logico. Se i volumi logici sono frammentati, l'output del suddetto comando sarà in grado di mostrarlo.

```
# lvs --segments
  LV          VG          Attr   #Str Type   SSize
LogVol100    VolGroup00 -wi-ao    1 linear 36.62G
LogVol101    VolGroup00 -wi-ao    1 linear 512.00M
lv           vg           -wi-a-    1 linear 104.00M
lv           vg           -wi-a-    1 linear 104.00M
lv           vg           -wi-a-    1 linear 104.00M
lv           vg           -wi-a-    1 linear 88.00M
```

Utilizzando **-v** con il comando **lvs --segments** aggiungerete i seguenti campi al display predefinito: **seg_start**, **stripesize**, **chunksize**.

```
# lvs -v --segments
    Finding all logical volumes
  LV          VG      Attr   Start SSize  #Str Type   Stripe Chunk
lv100        new_vg owi-a-    0 52.00M    1 linear    0    0
newvgsnap1   new_vg swi-a-    0 8.00M     1 linear    0 8.00K
```

Il seguente esempio mostra l'output predefinito di **lvs** su di un sistema con un volume logico configurato, seguito dall'output predefinito di **lvs** con **segments** specificato.

```
# lvs
  LV   VG      Attr   LSize  Origin Snap%  Move Log Copy%
lv100 new_vg -wi-a- 52.00M
# lvs --segments
  LV   VG      Attr   #Str Type   SSize
lv100 new_vg -wi-a-    1 linear 52.00M
```

4.8.3. Come ordinare i riporti di LVM

Normalmente l'intero output dei comandi **lvs**, **vgs**, o **pvs** deve essere generato e conservato internamente prima di poterlo ordinare ed eseguire un allineamento corretto delle colonne. Specificate l'opzione **--unbuffered** per visualizzare gli output non ordinati appena generati.

Per specificare un elenco ordinato alternativo di colonne sulle quali eseguire l'ordine utilizzate l'opzione **-O** di uno qualsiasi dei comandi usati per il riporto. Non è necessario includere questi campi all'interno dell'output.

Il seguente esempio mostra l'output di **pvs** il quale mostra il nome, la dimensione e lo spazio disponibile del volume fisico.

```
# pvs -o pv_name,pv_size,pv_free
PV          PSize  PFree
/dev/sdb1   17.14G 17.14G
/dev/sdc1   17.14G 17.09G
/dev/sdd1   17.14G 17.14G
```

Il seguente esempio mostra lo stesso output ordinato in base al campo dello spazio disponibile.

```
# pvs -o pv_name,pv_size,pv_free -O pv_free
PV          PSize  PFree
/dev/sdc1   17.14G 17.09G
/dev/sdd1   17.14G 17.14G
/dev/sdb1   17.14G 17.14G
```

Il seguente esempio mostra che non è necessario visualizzare il campo sul quale state eseguendo il riordino.

```
# pvs -o pv_name,pv_size -O pv_free
PV          PSize
/dev/sdc1   17.14G
/dev/sdd1   17.14G
/dev/sdb1   17.14G
```

Per visualizzare un processo di ordine inverso, aggiungete il segno **-** subito dopo il campo che contiene l'opzione **-O**.

```
# pvs -o pv_name,pv_size,pv_free -O -pv_free
PV          PSize  PFree
/dev/sdd1   17.14G 17.14G
/dev/sdb1   17.14G 17.14G
/dev/sdc1   17.14G 17.09G
```

4.8.4. Come specificare le unità

Per specificare le unità per il display di riporto LVM utilizzate l'opzione **--units**. È possibile specificare (b)ytes, (k)ilobytes, (m)egabytes, (g)igabytes, (t)erabytes, (e)xabytes, (p)etabytes, e (h)uman-readable. L'impostazione predefinita è human-readable. Potrete sovrascrivere l'impostazione predefinita impostando il parametro **units** nella sezione **global** del file **lvm.conf**.

Il seguente esempio specifica l'output di **pvs** in megabyte e non in gigabyte.

```
# pvs --units m
```

```

PV          VG      Fmt  Attr PSize      PFree
/dev/sda1   VG      lvm2 --   17555.40M 17555.40M
/dev/sdb1   new_vg  lvm2 a-   17552.00M 17552.00M
/dev/sdc1   new_vg  lvm2 a-   17552.00M 17500.00M
/dev/sdd1   new_vg  lvm2 a-   17552.00M 17552.00M

```

Per default le unità vengono visualizzate moltiplicate per 2 (multipli di 1024). È possibile specificare la visualizzazione delle unità in multipli di 1000, usando le specifiche dell'unità con la lettera maiuscola (B, K, M, G, T, H).

Il seguente comando mostra l'output come multiplo di 1024, comportamento predefinito.

```

# pvs
PV          VG      Fmt  Attr PSize  PFree
/dev/sdb1   new_vg  lvm2 a-   17.14G 17.14G
/dev/sdc1   new_vg  lvm2 a-   17.14G 17.09G
/dev/sdd1   new_vg  lvm2 a-   17.14G 17.14G

```

Il seguente comando mostra l'output come multiplo di 1000.

```

# pvs --units G
PV          VG      Fmt  Attr PSize  PFree
/dev/sdb1   new_vg  lvm2 a-   18.40G 18.40G
/dev/sdc1   new_vg  lvm2 a-   18.40G 18.35G
/dev/sdd1   new_vg  lvm2 a-   18.40G 18.40G

```

È possibile specificare (s)ectors 'settori' (definiti con 512 byte) oppure le unità personalizzate.

Il seguente esempio mostra l'output del comando **pvs** come un numero di settori.

```

# pvs --units s
PV          VG      Fmt  Attr PSize      PFree
/dev/sdb1   new_vg  lvm2 a-   35946496S 35946496S
/dev/sdc1   new_vg  lvm2 a-   35946496S 35840000S
/dev/sdd1   new_vg  lvm2 a-   35946496S 35946496S

```

Il seguente esempio mostra l'output del comando **pvs** in unità di 4 megabyte.

```

# pvs --units 4m
PV          VG      Fmt  Attr PSize      PFree
/dev/sdb1   new_vg  lvm2 a-   4388.00U 4388.00U
/dev/sdc1   new_vg  lvm2 a-   4388.00U 4375.00U
/dev/sdd1   new_vg  lvm2 a-   4388.00U 4388.00U

```

CAPITOLO 5. ESEMPI DI CONFIGURAZIONE LVM

Questo capitolo fornisce alcuni esempi di base per la configurazione di LVM.

5.1. CREAZIONE DI UN VOLUME LOGICO LVM SU TRE DISCHI

Questo esempio riporta la creazione di un volume logico LVM chiamato `new_logical_volume` il quale consiste nei dischi `/dev/sda1`, `/dev/sdb1`, e `/dev/sdc1`

5.1.1. Creazione dei volumi fisici

Per poter usare i dischi in un gruppo di volumi etichettateli come volumi fisici LVM.



AVVERTIMENTO

Questo comando distrugge qualsiasi dato presente su `/dev/sda1`, `/dev/sdb1`, e `/dev/sdc1`.

```
[root@tng3-1 ~]# pvcreate /dev/sda1 /dev/sdb1 /dev/sdc1
Physical volume "/dev/sda1" successfully created
Physical volume "/dev/sdb1" successfully created
Physical volume "/dev/sdc1" successfully created
```

5.1.2. Creazione del gruppo di volumi

Il seguente comando crea il gruppo di volumi `new_vol_group`.

```
[root@tng3-1 ~]# vgcreate new_vol_group /dev/sda1 /dev/sdb1 /dev/sdc1
Volume group "new_vol_group" successfully created
```

È possibile utilizzare il comando `vgs` per visualizzare gli attributi del nuovo gruppo di volumi.

```
[root@tng3-1 ~]# vgs
VG                #PV #LV #SN Attr   VSize  VFree
new_vol_group     3   0   0 wz--n- 51.45G 51.45G
```

5.1.3. Creazione del volume logico

Il seguente comando crea il volume logico `new_logical_volume` dal gruppo di volumi `new_vol_group`. In questo esempio viene creato un volume logico che utilizza 2 GB del gruppo di volumi.

```
[root@tng3-1 ~]# lvcreate -L2G -n new_logical_volume new_vol_group
Logical volume "new_logical_volume" created
```

5.1.4. Creazione del file system

Il seguente comando crea un file system GFS2 sul volume logico.

```
[root@tng3-1 ~]# mkfs.gfs2 -plock_nolock -j 1
/dev/new_vol_group/new_logical_volume
This will destroy any data on /dev/new_vol_group/new_logical_volume.

Are you sure you want to proceed? [y/n] y

Device:                /dev/new_vol_group/new_logical_volume
Blocksize:             4096
Filesystem Size:       491460
Journals:              1
Resource Groups:       8
Locking Protocol:      lock_nolock
Lock Table:

Syncing...
All Done
```

I seguenti comandi montano il volume logico e riportano l'utilizzo dello spazio del disco del file system.

```
[root@tng3-1 ~]# mount /dev/new_vol_group/new_logical_volume /mnt
[root@tng3-1 ~]# df
Filesystem                1K-blocks      Used Available Use% Mounted on
/dev/new_vol_group/new_logical_volume
                        1965840         20   1965820    1% /mnt
```

5.2. CREAZIONE DI UN VOLUME LOGICO SEGMENTATO

In questo esempio viene creato un volume logico segmentato LVM chiamato `striped_logical_volume` il quale scrive i dati sui dischi `/dev/sda1`, `/dev/sdb1`, e `/dev/sdc1`.

5.2.1. Creazione dei volumi fisici

Etichettare i dischi da utilizzare nei gruppi di volumi come volumi fisici LVM.



AVVERTIMENTO

Questo comando distrugge qualsiasi dato presente su `/dev/sda1`, `/dev/sdb1`, e `/dev/sdc1`.

```
[root@tng3-1 ~]# pvcreate /dev/sda1 /dev/sdb1 /dev/sdc1
Physical volume "/dev/sda1" successfully created
Physical volume "/dev/sdb1" successfully created
Physical volume "/dev/sdc1" successfully created
```

5.2.2. Creazione del gruppo di volumi

Il seguente comando crea il gruppo di volumi **volgroup01**.

```
[root@tng3-1 ~]# vgcreate volgroup01 /dev/sda1 /dev/sdb1 /dev/sdc1
Volume group "volgroup01" successfully created
```

È possibile utilizzare il comando **vgs** per visualizzare gli attributi del nuovo gruppo di volumi.

```
[root@tng3-1 ~]# vgs
VG                #PV #LV #SN Attr   VSize  VFree
volgroup01        3   0   0 wz--n- 51.45G 51.45G
```

5.2.3. Creazione del volume logico

Il seguente comando crea il volume logico segmentato **striped_logical_volume** dal gruppo di volumi **volgroup01**. In questo esempio viene creato un volume logico con una misura di 2 gigabyte, con tre segmenti con una dimensione di 4 kilobyte l'una.

```
[root@tng3-1 ~]# lvcreate -i3 -I4 -L2G -nstriped_logical_volume volgroup01
Rounding size (512 extents) up to stripe boundary size (513 extents)
Logical volume "striped_logical_volume" created
```

5.2.4. Creazione del file system

Il seguente comando crea un file system GFS2 sul volume logico.

```
[root@tng3-1 ~]# mkfs.gfs2 -plock_nolock -j 1
/dev/volgroup01/striped_logical_volume
This will destroy any data on /dev/volgroup01/striped_logical_volume.
```

```
Are you sure you want to proceed? [y/n] y
```

```
Device:                /dev/volgroup01/striped_logical_volume
Blocksize:              4096
Filesystem Size:        492484
Journals:               1
Resource Groups:        8
Locking Protocol:       lock_nolock
Lock Table:
```

```
Syncing...
All Done
```

I seguenti comandi montano il volume logico e riportano l'utilizzo dello spazio del disco del file system.

```
[root@tng3-1 ~]# mount /dev/volgroup01/striped_logical_volume /mnt
[root@tng3-1 ~]# df
Filesystem              1K-blocks      Used Available Use% Mounted on
/dev/mapper/VolGroup00-LogVol00
                        13902624    1656776   11528232   13% /
/dev/hda1                101086      10787     85080    12% /boot
```

```

tmpfs                127880          0    127880    0% /dev/shm
/dev/volgroup01/striped_logical_volume
                    1969936          20   1969916    1% /mnt

```

5.3. SEPARAZIONE DI UN GRUPPO DI VOLUMI

In questo esempio un gruppo di volumi esistente consiste in tre volumi fisici. Se è disponibile spazio sufficiente non utilizzato sui volumi fisici, sarà possibile creare un nuovo gruppo di volumi senza aggiungere nuovi dischi.

Nell'impostazione iniziale il volume logico **mylv** viene ottenuto dal gruppo di volumi **myvol**, il quale a sua volta consiste di tre volumi fisici, **/dev/sda1**, **/dev/sdb1**, e **/dev/sdc1**.

Dopo aver completato questa procedura il gruppo di volumi **myvg** sarà composto da **/dev/sda1** e **/dev/sdb1**. Un secondo gruppo di volumi, **yourvg** sarà composta da **/dev/sdc1**.

5.3.1. Come determinare lo spazio disponibile

Per determinare la quantità di spazio attualmente disponibile in un gruppo di volumi, utilizzate il comando **pvscan**.

```

[root@tng3-1 ~]# pvscan
PV /dev/sda1  VG myvg   lvm2 [17.15 GB / 0    free]
PV /dev/sdb1  VG myvg   lvm2 [17.15 GB / 12.15 GB free]
PV /dev/sdc1  VG myvg   lvm2 [17.15 GB / 15.80 GB free]
Total: 3 [51.45 GB] / in use: 3 [51.45 GB] / in no VG: 0 [0    ]

```

5.3.2. Come spostare i dati

È possibile spostare tutte le estensioni fisiche presenti in **/dev/sdc1** su **/dev/sdb1** tramite **pvmove**. **pvmove** può richiedere un periodo molto lungo per la sua esecuzione.

```

[root@tng3-1 ~]# pvmove /dev/sdc1 /dev/sdb1
/dev/sdc1: Moved: 14.7%
/dev/sdc1: Moved: 30.3%
/dev/sdc1: Moved: 45.7%
/dev/sdc1: Moved: 61.0%
/dev/sdc1: Moved: 76.6%
/dev/sdc1: Moved: 92.2%
/dev/sdc1: Moved: 100.0%

```

Dopo aver spostato i dati potrete notare che tutto lo spazio su **/dev/sdc1** è ora disponibile.

```

[root@tng3-1 ~]# pvscan
PV /dev/sda1  VG myvg   lvm2 [17.15 GB / 0    free]
PV /dev/sdb1  VG myvg   lvm2 [17.15 GB / 10.80 GB free]
PV /dev/sdc1  VG myvg   lvm2 [17.15 GB / 17.15 GB free]
Total: 3 [51.45 GB] / in use: 3 [51.45 GB] / in no VG: 0 [0    ]

```

5.3.3. Come dividere il gruppo di volumi

Per creare un nuovo gruppo di volumi **yourvg**, utilizzate il comando **vgsplit** per dividere il gruppo di volumi **myvg**.

Prima di poter dividere il gruppo di volumi, il volume logico deve essere inattivo. Se il file system risulta montato sarà necessario smontarlo prima di rendere il volume logico inattivo.

È possibile rendere inattivi i volumi logici con il comando **lvchange** o **vgchange**. Il seguente comando rende inattivo il volume logico **mylv**, e successivamente divide il gruppo di volumi **yourvg** dal gruppo di volumi **myvg**, spostando il volume fisico **/dev/sdc1** nel nuovo gruppo di volumi **yourvg**.

```
[root@tng3-1 ~]# lvchange -a n /dev/myvg/mylv
[root@tng3-1 ~]# vgsplit myvg yourvg /dev/sdc1
Volume group "yourvg" successfully split from "myvg"
```

Utilizzate **vgs** per visualizzare gli attributi dei due gruppi di volumi.

```
[root@tng3-1 ~]# vgs
VG      #PV #LV #SN Attr   VSize  VFree
myvg    2   1   0 wz--n- 34.30G 10.80G
yourvg  1   0   0 wz--n- 17.15G 17.15G
```

5.3.4. Creazione di un nuovo volume logico

Dopo aver creato un nuovo gruppo di volumi sarà possibile creare il nuovo volume logico **yourlv**.

```
[root@tng3-1 ~]# lvcreate -L5G -n yourlv yourvg
Logical volume "yourlv" created
```

5.3.5. Creazione di un file system e montaggio di un nuovo volume logico

È possibile creare un file system su di un nuovo volume logico e successivamente montarlo.

```
[root@tng3-1 ~]# mkfs.gfs2 -plock_nolock -j 1 /dev/yourvg/yourlv
This will destroy any data on /dev/yourvg/yourlv.
```

```
Are you sure you want to proceed? [y/n] y
```

```
Device:                /dev/yourvg/yourlv
Blocksize:              4096
Filesystem Size:        1277816
Journals:               1
Resource Groups:        20
Locking Protocol:       lock_nolock
Lock Table:
```

```
Syncing...
All Done
```

```
[root@tng3-1 ~]# mount /dev/yourvg/yourlv /mnt
```

5.3.6. Attivazione e montaggio del volume logico originale

Poichè è necessario rendere inattivo il volume logico **mylv**, sarà necessario riattivarlo prima di poterlo montare.

```
root@tng3-1 ~]# lvchange -a y mylv

[root@tng3-1 ~]# mount /dev/myvg/mylv /mnt
[root@tng3-1 ~]# df
Filesystem                1K-blocks      Used Available Use% Mounted on
/dev/yourvg/yourlv         24507776         32   24507744   1% /mnt
/dev/myvg/mylv             24507776         32   24507744   1% /mnt
```

5.4. RIMOZIONE DI UN DISCO DA UN VOLUME LOGICO

Questo esempio mostra come rimuovere un disco da un volume logico esistente sia per sostituire il disco che per il suo utilizzo come parte di un volume diverso. Per poter rimuovere un disco, è necessario spostare le estensioni del volume fisico LVM su di un disco diverso o set di dischi.

5.4.1. Come spostare le estensioni su volumi fisici esistenti

In questo esempio il volume logico è distribuito attraverso quattro volumi fisici nel gruppo di volumi **myvg**.

```
[root@tng3-1]# pvs -o+pv_used
PV          VG   Fmt Attr PSize  PFree  Used
/dev/sda1   myvg lvm2 a-   17.15G 12.15G  5.00G
/dev/sdb1   myvg lvm2 a-   17.15G 12.15G  5.00G
/dev/sdc1   myvg lvm2 a-   17.15G 12.15G  5.00G
/dev/sdd1   myvg lvm2 a-   17.15G  2.15G 15.00G
```

Spostiamo le estensioni fuori da **/dev/sdb1** in modo da poter rimuoverlo dal gruppo di volumi.

Se sono disponibili un numero di estensioni sufficienti su altri volumi fisici nel gruppo di volumi, sarà possibile eseguire il comando **pvmove** senza alcuna opzione sul dispositivo che desiderate rimuovere, così facendo le estensioni verranno distribuite su altri dispositivi.

```
[root@tng3-1 ~]# pvmove /dev/sdb1
/dev/sdb1: Moved: 2.0%
...
/dev/sdb1: Moved: 79.2%
...
/dev/sdb1: Moved: 100.0%
```

Dopo l'esecuzione del comando **pvmove**, la distribuzione delle estensioni risulta essere:

```
[root@tng3-1]# pvs -o+pv_used
PV          VG   Fmt Attr PSize  PFree  Used
/dev/sda1   myvg lvm2 a-   17.15G  7.15G 10.00G
/dev/sdb1   myvg lvm2 a-   17.15G 17.15G    0
/dev/sdc1   myvg lvm2 a-   17.15G 12.15G  5.00G
/dev/sdd1   myvg lvm2 a-   17.15G  2.15G 15.00G
```

Utilizzate il comando **vgreduce** per rimuovere il physical volume **/dev/sdb1** dal gruppo di volumi.

```
[root@tng3-1 ~]# vgreduce myvg /dev/sdb1
Removed "/dev/sdb1" from volume group "myvg"
[root@tng3-1 ~]# pvs
PV          VG   Fmt  Attr PSize  PFree
/dev/sda1   myvg lvm2 a-   17.15G  7.15G
/dev/sdb1    myvg lvm2 --   17.15G 17.15G
/dev/sdc1   myvg lvm2 a-   17.15G 12.15G
/dev/sdd1   myvg lvm2 a-   17.15G  2.15G
```

Ora il disco potrà essere rimosso fisicamente o assegnato ad altri utenti.

5.4.2. Come spostare le estensioni su di un nuovo disco

In questo esempio il volume logico viene distribuito attraverso i tre volumi fisici nel gruppo di volumi **myvg** nel modo seguente:

```
[root@tng3-1]# pvs -o+pv_used
PV          VG   Fmt  Attr PSize  PFree  Used
/dev/sda1   myvg lvm2 a-   17.15G  7.15G 10.00G
/dev/sdb1   myvg lvm2 a-   17.15G 15.15G  2.00G
/dev/sdc1   myvg lvm2 a-   17.15G 15.15G  2.00G
```

Desideriamo spostare le estensioni di **/dev/sdb1** su di un nuovo dispositivo, **/dev/sdd1**.

5.4.2.1. Creazione di un nuovo Physical Volume

Create un nuovo volume fisico da **/dev/sdd1**.

```
[root@tng3-1 ~]# pvcreate /dev/sdd1
Physical volume "/dev/sdd1" successfully created
```

5.4.2.2. Aggiungete il nuovo Physical Volume al gruppo di volumi

Aggiungete **/dev/sdd1** sul gruppo di volumi esistente **myvg**.

```
[root@tng3-1 ~]# vgextend myvg /dev/sdd1
Volume group "myvg" successfully extended
[root@tng3-1]# pvs -o+pv_used
PV          VG   Fmt  Attr PSize  PFree  Used
/dev/sda1   myvg lvm2 a-   17.15G  7.15G 10.00G
/dev/sdb1   myvg lvm2 a-   17.15G 15.15G  2.00G
/dev/sdc1   myvg lvm2 a-   17.15G 15.15G  2.00G
/dev/sdd1   myvg lvm2 a-   17.15G 17.15G    0
```

5.4.2.3. Come spostare i dati

Usate **pvmove** per spostare i dati da **/dev/sdb1** a **/dev/sdd1**.

```
[root@tng3-1 ~]# pvmove /dev/sdb1 /dev/sdd1
/dev/sdb1: Moved: 10.0%
...
/dev/sdb1: Moved: 79.7%
```

```
...
/dev/sdb1: Moved: 100.0%

[root@tng3-1]# pvs -o+pv_used
PV          VG   Fmt  Attr PSize  PFree  Used
/dev/sda1   myvg lvm2 a-   17.15G  7.15G 10.00G
/dev/sdb1   myvg lvm2 a-   17.15G 17.15G   0
/dev/sdc1   myvg lvm2 a-   17.15G 15.15G  2.00G
/dev/sdd1   myvg lvm2 a-   17.15G 15.15G  2.00G
```

5.4.2.4. Rimozione di un volume fisico vecchio dal gruppo di volumi

Dopo aver spostato i dati da `/dev/sdb1`, potrete eseguire la sua rimozione dal gruppo di volumi.

```
[root@tng3-1 ~]# vgreduce myvg /dev/sdb1
Removed "/dev/sdb1" from volume group "myvg"
```

Ora è possibile riassegnare il disco ad un altro gruppo di volumi o rimuoverlo dal sistema.

5.5. CREAZIONE DI UN VOLUME LOGICO LVM SPECULARE IN UN CLUSTER

La creazione di un volume logico LVM speculare in un cluster ha bisogno di comandi e procedure simili usati nella creazione di un volume logico LVM speculare su di un nodo. Tuttavia per creare un volume LVM speculare in un cluster, il cluster e l'infrastruttura del cluster mirror devono essere in esecuzione, il cluster deve essere quorato ed il tipo di blocco nel file `lvm.conf` deve essere impostato correttamente in modo da abilitare il blocco del cluster sia direttamente che per mezzo del comando `lvmconf` come descritto in [Sezione 3.1, «Creazione dei volumi LVM in un cluster»](#).

La seguente procedura è in grado di creare un volume LVM speculare in un cluster. Come prima cosa la procedura controlla se i servizi del cluster sono stati impostati ed in esecuzione, successivamente verrà creato il volume speculare.

1. Per creare un volume logico speculare condiviso da tutti i nodi presenti in un cluster il tipo di blocco deve essere impostato correttamente nel file `lvm.conf` su ogni nodo presente nel cluster. Per impostazione predefinita il tipo di blocco è impostato su locale. Per modificarlo eseguire il seguente comando in ogni nodo del cluster per abilitare il blocco clusterizzato:

```
# /sbin/lvmconf --enable-cluster
```

2. Per creare un volume logico clusterizzato l'infrastruttura del cluster deve essere in esecuzione su ogni nodo presente nel cluster. Il seguente esempio verifica che il demone `clvmd` è in esecuzione sul nodo dal quale è stato emesso:

```
[root@doc-07 ~]# ps auxw | grep clvmd
root      17642  0.0  0.1 32164 1072 ?        Ssl  Apr06   0:00
clvmd -T20 -t 90
```

Il seguente comando mostra la visuale locale dello stato del cluster:

```
[root@example-01 ~]# cman_tool services
fence domain
member count 3
```

```

victim count 0
victim now 0
master nodeid 2
wait state none
members 1 2 3

dml lockspaces
name clvmd
id 0x4104eefa
flags 0x00000000
change member 3 joined 1 remove 0 failed 0 seq 1,1
members 1 2 3

```

3. Assicurarsi che il pacchetto **cmirror** sia stato installato.

4. Avviare il servizio **cmirrord**.

```

[root@hexample-01 ~]# service cmirrord start
Starting cmirrord: [ OK
]

```

5. Creare il mirror. Creare prima i volumi fisici. I seguenti comandi creeranno tre volumi fisici. Due dei volumi fisici verranno usati per le 'gambe' o sezioni del mirror ed il terzo conterrà il log del mirror.

```

[root@doc-07 ~]# pvcreate /dev/xvdb1
Physical volume "/dev/xvdb1" successfully created
[root@doc-07 ~]# pvcreate /dev/xvdb2
Physical volume "/dev/xvdb2" successfully created
[root@doc-07 ~]# pvcreate /dev/xvdc1
Physical volume "/dev/xvdc1" successfully created

```

6. Creare il gruppo di volumi. In questo esempio viene creato un gruppo di volumi **vg001** il quale consiste in tre volumi fisici creati nella fase precedente.

```

[root@doc-07 ~]# vgcreate vg001 /dev/xvdb1 /dev/xvdb2 /dev/xvdc1
Clustered volume group "vg001" successfully created

```

Da notare che l'output del comando **vgcreate** indica che il gruppo di volumi è clusterizzato. Per verificare se un gruppo di volumi è clusterizzato usare il comando **vgs**, il quale mostrerà gli attributi del gruppo di volumi. Se un gruppo di volumi è clusterizzato verrà mostrato un attributo **c**.

```

[root@doc-07 ~]# vgs vg001
VG          #PV #LV #SN Attr   VSize  VFree
vg001      3   0   0 wz--nc 68.97G 68.97G

```

7. Creare il volume logico segmentato. In questo esempio è stato creato il volume logico **mirrorlv** dal gruppo di volumi **vg001**. Questo volume presenta una sezione del mirror. In questo esempio viene specificato quale estensione del volume fisico verrà usata per il volume logico.

```
[root@doc-07 ~]# lvcreate -l 1000 -m1 vg001 -n mirrorlv
/dev/xvdb1:1-1000 /dev/xvdb2:1-1000 /dev/xvdc1:0
Logical volume "mirrorlv" created
```

Usare il comando **lvs** per mostrare i progressi della creazione del mirror. Il seguente esempio mostra che il mirror è al 47%, successivamente al 91% e subito dopo al 100% quando il mirror è completo.

```
[root@doc-07 log]# lvs vg001/mirrorlv
LV          VG          Attr      LSize Origin Snap%   Move Log
Copy%      Convert
mirrorlv    vg001          mwi-a-  3.91G                        vg001_mlog
47.00
[root@doc-07 log]# lvs vg001/mirrorlv
LV          VG          Attr      LSize Origin Snap%   Move Log
Copy%      Convert
mirrorlv    vg001          mwi-a-  3.91G                        vg001_mlog
91.00
[root@doc-07 ~]# lvs vg001/mirrorlv
LV          VG          Attr      LSize Origin Snap%   Move Log
Copy%      Convert
mirrorlv    vg001          mwi-a-  3.91G                        vg001_mlog
100.00
```

Il completamento del processo viene riportato nel log del sistema:


```
May 10 14:52:52 doc-07 [19402]: Monitoring mirror device vg001-
mirrorlv for events
May 10 14:55:00 doc-07 lvm[19402]: vg001-mirrorlv is now in-sync
```

8. Usare **lvs** con l'opzione **-o +devices** per visualizzare la configurazione ed il dispositivo che rappresenta la 'gamba' del mirror. In questo esempio è possibile vedere come il volume logico è composto da due immagini lineari ed un log.

```
[root@doc-07 ~]# lvs -a -o +devices
LV          VG          Attr      LSize  Origin Snap%   Move
Log          Copy%      Convert Devices
mirrorlv          vg001          mwi-a-  3.91G
mirrorlv_mlog 100.00
mirrorlv_mimage_0(0),mirrorlv_mimage_1(0)
[mirrorlv_mimage_0] vg001          iwi-ao  3.91G
/dev/xvdb1(1)
[mirrorlv_mimage_1] vg001          iwi-ao  3.91G
/dev/xvdb2(1)
[mirrorlv_mlog]    vg001          lwi-ao  4.00M
/dev/xvdc1(0)
```

Usare l'opzione **seg_pe_ranges** del comando **lvs** per visualizzare il layout dei dati. È possibile usare questa opzione per verificare la ridondanza corretta del layout. L'output di questo comando mostra le gamme di PE con lo stesso formato accettato come input da **lvcreate** e **lvresize**.

```
[root@doc-07 ~]# lvs -a -o +seg_pe_ranges --segments
PE Ranges
```



```
mirrorlv_mimage_0:0-999 mirrorlv_mimage_1:0-999
/dev/xvdb1:1-1000
/dev/xvdb2:1-1000
/dev/xvdc1:0-0
```



NOTA

Per informazioni sul ripristino da un errore di uno dei segmenti di un volume speculare LVM consultare [Sezione 6.3, «Processo di recupero da un LVM Mirror Failure»](#) .

CAPITOLO 6. TROUBLESHOOTING DI LVM

Questo capitolo fornisce le informazioni necessarie per il troubleshooting di una varietà di problemi relativi a LVM.

6.1. INFORMAZIONI DIAGNOSTICHE PER IL TROUBLESHOOTING

Se un comando non funziona come previsto sarà possibile ottenere informazioni diagnostiche nei seguenti modi:

- Utilizzate le opzioni **-v**, **-vv**, **-vvv**, o **-vvvv** di qualsiasi comando, per livelli sempre più verbosi dell'output.
- Se il problema è dovuto all'attivazione del volume logico, impostate **'activation = 1'** nella sezione **'log'** del file di configurazione ed eseguite il comando con l'opzione **-vvvv**. Dopo aver completato il controllo dell'output assicuratevi di resettare questo parametro su 0, in modo da evitare possibili problemi con il lock della macchina durante situazioni in cui avrete una memoria molto bassa.
- Eseguite il comando **lvm dump** il quale fornisce le informazioni relative al dump per scopi diagnostici. Per informazioni consultate la pagina man (8) di **lvm dump**.
- Eseguite il comando **lvs -v**, **pvs -a** o **dmsetup info -c** per informazioni aggiuntive del sistema.
- Esaminate l'ultimo backup dei metadati in **/etc/lvm/backup** e le versioni archiviate in **/etc/lvm/archive**.
- Controllate le informazioni correnti sulla configurazione, eseguendo il comando **lvm dumpconfig**.
- Controllate il file **.cache** in **/etc/lvm** per informazioni relative sul dispositivo che presenta i volumi fisici.

6.2. COME VISUALIZZARE LE INFORMAZIONI SU DISPOSITIVI FALLITI

È possibile usare l'opzione **-P** del comando **lvs** o **vgs**, per visualizzare le informazioni sul volume fallito che potrebbero non apparire all'interno dell'output. Questa opzione permette alcune operazioni anche se i metadati non sono completamente uguali all'interno. Per esempio, se uno dei dispositivi che compone il gruppo di volumi **vg** fallisce, il comando **vgs** potrebbe mostrare il seguente output.

```
[root@link-07 tmp]# vgs -o +devices
Volume group "vg" not found
```

Se specificate l'opzione **-P** di **vgs**, il gruppo di volumi risulta essere ancora non utilizzabile, ma sarete in grado di visualizzare maggiori informazioni sul dispositivo fallito.

```
[root@link-07 tmp]# vgs -P -o +devices
Partial mode. Incomplete volume groups will be activated read-only.
VG    #PV #LV #SN Attr   VSize VFree Devices
vg     9  2   0 rz-pn- 2.11T 2.07T unknown device(0)
vg     9  2   0 rz-pn- 2.11T 2.07T unknown device(5120),/dev/sda1(0)
```

In questo esempio il dispositivo fallito ha causato il fallimento del volume logico segmentato e lineare. **lvs** senza l'opzione **-P** mostra il seguente output.

```
[root@link-07 tmp]# lvs -a -o +devices
Volume group "vg" not found
```

Utilizzando l'opzione **-P** visualizzerete i volumi logici falliti.

```
[root@link-07 tmp]# lvs -P -a -o +devices
Partial mode. Incomplete volume groups will be activated read-only.
LV      VG      Attr      LSize   Origin Snap%   Move Log Copy%  Devices
linear  vg      -wi-a-    20.00G                                     unknown
device(0)
stripe  vg      -wi-a-    20.00G                                     unknown
device(5120),/dev/sda1(0)
```

I seguenti esempi mostrano l'output dei comandi **pvs** e **lvs** con l'opzione **-P** specificata quando una sezione del volume logico speculare è fallita.

```
root@link-08 ~]# vgs -a -o +devices -P
Partial mode. Incomplete volume groups will be activated read-only.
VG      #PV #LV #SN Attr   VSize VFree Devices
corey   4   4   0 rz-pnc 1.58T 1.34T
my_mirror_mimage_0(0),my_mirror_mimage_1(0)
corey   4   4   0 rz-pnc 1.58T 1.34T /dev/sdd1(0)
corey   4   4   0 rz-pnc 1.58T 1.34T unknown device(0)
corey   4   4   0 rz-pnc 1.58T 1.34T /dev/sdb1(0)
```

```
[root@link-08 ~]# lvs -a -o +devices -P
Partial mode. Incomplete volume groups will be activated read-only.
LV      VG      Attr      LSize   Origin Snap%   Move Log Copy%  Devices
my_mirror      corey mwi-a-    120.00G
my_mirror_mlog 1.95 my_mirror_mimage_0(0),my_mirror_mimage_1(0)
[my_mirror_mimage_0] corey iwi-ao    120.00G
unknown device(0)
[my_mirror_mimage_1] corey iwi-ao    120.00G
/dev/sdb1(0)
[my_mirror_mlog]   corey lwi-ao      4.00M
/dev/sdd1(0)
```

6.3. PROCESSO DI RECUPERO DA UN LVM MIRROR FAILURE

Questa sezione fornisce un esempio sul ripristino da un fallimento di una sezione del volume speculare LVM a causa del fallimento del dispositivo sottostante per un volume fisico, con il parametro **mirror_log_fault_policy** impostato su **remove** rendendo necessaria una compilazione manuale del mirror. Per maggiori informazioni su come impostare il parametro **mirror_log_fault_policy** consultare [Sezione 6.3, «Processo di recupero da un LVM Mirror Failure»](#).

Quando una sezione del mirror fallisce LVM converte il volume speculare in volume lineare il quale a sua volta continua ad operare senza però avere una ridondanza speculare. A questo punto è possibile aggiungere un nuovo dispositivo a disco sul sistema da utilizzare come dispositivo fisico di sostituzione e successivamente ricompilare il mirror.

Il seguente comando crea i volumi fisici che verranno utilizzati per il mirror.

```
[root@link-08 ~]# pvcreate /dev/sd[abcdefgh][12]
Physical volume "/dev/sda1" successfully created
Physical volume "/dev/sda2" successfully created
Physical volume "/dev/sdb1" successfully created
Physical volume "/dev/sdb2" successfully created
Physical volume "/dev/sdc1" successfully created
Physical volume "/dev/sdc2" successfully created
Physical volume "/dev/sdd1" successfully created
Physical volume "/dev/sdd2" successfully created
Physical volume "/dev/sde1" successfully created
Physical volume "/dev/sde2" successfully created
Physical volume "/dev/sdf1" successfully created
Physical volume "/dev/sdf2" successfully created
Physical volume "/dev/sdg1" successfully created
Physical volume "/dev/sdg2" successfully created
Physical volume "/dev/sdh1" successfully created
Physical volume "/dev/sdh2" successfully created
```

I seguenti comandi creano il gruppo di volumi 'volume group' **vg** ed i volumi speculari **groupfs**.

```
[root@link-08 ~]# vgcreate vg /dev/sd[abcdefgh][12]
Volume group "vg" successfully created
[root@link-08 ~]# lvcreate -L 750M -n groupfs -m 1 vg /dev/sda1 /dev/sdb1
/dev/sdc1
Rounding up size to full physical extent 752.00 MB
Logical volume "groupfs" created
```

È possibile utilizzare il comando **lvs** per verificare la disposizione del volume speculare e dei dispositivi relativi alla sezione del mirror e del log. Da notare che nel primo esempio il mirror non è ancora completamente sincronizzato; dovrete attendere fino a quando il campo **Copy%** mostra il valore 100.00 prima di continuare.

```
[root@link-08 ~]# lvs -a -o +devices
LV          VG      Attr      LSize    Origin Snap%   Move Log
Copy% Devices
groupfs      vg      mwi-a-    752.00M                                groupfs_mlog
21.28 groupfs_mimage_0(0),groupfs_mimage_1(0)
[groupfs_mimage_0] vg      iwi-ao    752.00M
/dev/sda1(0)
[groupfs_mimage_1] vg      iwi-ao    752.00M
/dev/sdb1(0)
[groupfs_mlog]   vg      lwi-ao     4.00M
/dev/sdc1(0)

[root@link-08 ~]# lvs -a -o +devices
LV          VG      Attr      LSize    Origin Snap%   Move Log
Copy% Devices
groupfs      vg      mwi-a-    752.00M                                groupfs_mlog
100.00 groupfs_mimage_0(0),groupfs_mimage_1(0)
[groupfs_mimage_0] vg      iwi-ao    752.00M
/dev/sda1(0)
[groupfs_mimage_1] vg      iwi-ao    752.00M
```

```
/dev/sdb1(0)
[groupfs_mlog]    vg    lwi-ao    4.00M    i
/dev/sdc1(0)
```

In questo esempio la sezione primaria del mirror **/dev/sda1** fallisce. Ogni attività di scrittura sul volume speculare causa il rilevamento da parte di LVM del mirror fallito. Quando ciò accade, LVM converte il mirror in un volume lineare singolo. In questo caso per attivare la conversione, verrà eseguito un comando **dd**

```
[root@link-08 ~]# dd if=/dev/zero of=/dev/vg/groupfs count=10
10+0 records in
10+0 records out
```

Usare il comando **lvs** per verificare se il dispositivo è un dispositivo lineare. A causa del fallimento del disco si verificheranno errori I/O.

```
[root@link-08 ~]# lvs -a -o +devices
/dev/sda1: read failed after 0 of 2048 at 0: Input/output error
/dev/sda2: read failed after 0 of 2048 at 0: Input/output error
LV      VG    Attr   LSize   Origin Snap%   Move Log Copy%  Devices
groupfs vg    -wi-a- 752.00M                               /dev/sdb1(0)
```

A questo punto dovrete essere in grado di usare il volume logico senza però avere una ridondanza del mirror.

Per ricompilare il volume speculare sostituite l'unità non funzionante e ricreate il volume fisico. Se utilizzate lo stesso disco invece di sostituirlo con un disco nuovo visualizzerete alcuni avvertimenti "inconsistenti" quando eseguirete il comando **pvcreeate**. Per non visualizzare questo messaggio eseguire **vgreduce --removemissing**.

```
[root@link-08 ~]# pvcreate /dev/sdi[12]
Physical volume "/dev/sdi1" successfully created
Physical volume "/dev/sdi2" successfully created

[root@link-08 ~]# pvscan
PV /dev/sdb1    VG vg    lvm2 [67.83 GB / 67.10 GB free]
PV /dev/sdb2    VG vg    lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdc1    VG vg    lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdc2    VG vg    lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdd1    VG vg    lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdd2    VG vg    lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sde1    VG vg    lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sde2    VG vg    lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdf1    VG vg    lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdf2    VG vg    lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdg1    VG vg    lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdg2    VG vg    lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdh1    VG vg    lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdh2    VG vg    lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdi1    VG vg    lvm2 [603.94 GB]
PV /dev/sdi2    VG vg    lvm2 [603.94 GB]
Total: 16 [2.11 TB] / in use: 14 [949.65 GB] / in no VG: 2 [1.18 TB]
```

Successivamente estendete il gruppo di volumi originale con il nuovo volume fisico.

```
[root@link-08 ~]# vgextend vg /dev/sdi[12]
Volume group "vg" successfully extended

[root@link-08 ~]# pvscan
PV /dev/sdb1    VG vg    lvm2 [67.83 GB / 67.10 GB free]
PV /dev/sdb2    VG vg    lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdc1    VG vg    lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdc2    VG vg    lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdd1    VG vg    lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdd2    VG vg    lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sde1    VG vg    lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sde2    VG vg    lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdf1    VG vg    lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdf2    VG vg    lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdg1    VG vg    lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdg2    VG vg    lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdh1    VG vg    lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdh2    VG vg    lvm2 [67.83 GB / 67.83 GB free]
PV /dev/sdi1    VG vg    lvm2 [603.93 GB / 603.93 GB free]
PV /dev/sdi2    VG vg    lvm2 [603.93 GB / 603.93 GB free]
Total: 16 [2.11 TB] / in use: 16 [2.11 TB] / in no VG: 0 [0  ]
```

Convertite il volume lineare nel suo stato speculare originale.

```
[root@link-08 ~]# lvconvert -m 1 /dev/vg/groupfs /dev/sdi1 /dev/sdb1
/dev/sdc1
Logical volume mirror converted.
```

Potrete utilizzare il comando `lvs` per verificare che il mirror sia stato ripristinato.

```
[root@link-08 ~]# lvs -a -o +devices
LV          VG   Attr   LSize   Origin Snap%  Move Log
Copy% Devices
groupfs      vg   mwi-a- 752.00M                      groupfs_mlog
68.62 groupfs_mimage_0(0),groupfs_mimage_1(0)
[groupfs_mimage_0] vg   iwi-ao 752.00M
/dev/sdb1(0)
[groupfs_mimage_1] vg   iwi-ao 752.00M
/dev/sdi1(0)
[groupfs_mlog]   vg   lwi-ao  4.00M
/dev/sdc1(0)
```

6.4. RECUPERO DEI METADATI DEL VOLUME FISICO

Se l'area dei metadati del gruppo di volumi di un volume fisico viene accidentalmente sovrascritta o distrutta, sarà possibile visualizzare un messaggio d'errore il quale indica che l'area interessata è incorretta, o che il sistema non è stato in grado di trovare un volume fisico con un UUID particolare. È possibile recuperare i dati del volume fisico creando una nuova area per i metadati sul volume fisico stesso specificando un UUID uguale a quello dei metadati precedentemente persi.



AVVERTIMENTO

Non eseguite questa procedura con un volume logico LVM in funzione. In tal caso potreste perdere i vostri dati se specificate l'UUID incorretto.

Il seguente esempio mostra un tipo di output che potreste visualizzare se l'area dei metadati viene persa o se risulta corrotta.

```
[root@link-07 backup]# lvs -a -o +devices
Couldn't find device with uuid 'FmGRh3-zhok-iVI8-7qTD-S5BI-MAEN-NYM5Sk'.
Couldn't find all physical volumes for volume group VG.
Couldn't find device with uuid 'FmGRh3-zhok-iVI8-7qTD-S5BI-MAEN-NYM5Sk'.
Couldn't find all physical volumes for volume group VG.
...
```

È possibile localizzare l'UUID per il volume fisico sovrascritto controllando la directory `/etc/lvm/archive`. Verificate il file `VolumeGroupName_xxxx.vg` per gli ultimi metadati di LVM validi archiviati, per quel gruppo di volumi.

Alternativamente la disattivazione del volume e l'impostazione dell'opzione **partial** (`-P`), potrebbe permettervi di trovare l'UUID del volume fisico corrotto mancante.

```
[root@link-07 backup]# vgchange -an --partial
Partial mode. Incomplete volume groups will be activated read-only.
Couldn't find device with uuid 'FmGRh3-zhok-iVI8-7qTD-S5BI-MAEN-NYM5Sk'.
Couldn't find device with uuid 'FmGRh3-zhok-iVI8-7qTD-S5BI-MAEN-NYM5Sk'.
...
```

Utilizzate le opzioni `--uuid` e `--restorefile` di `pvcreate` per ripristinare il volume fisico. Il seguente esempio etichetta il dispositivo `/dev/sdh1` come volume fisico con il seguente UUID, **FmGRh3-zhok-iVI8-7qTD-S5BI-MAEN-NYM5Sk**. Questo comando ripristina le informazioni relative ai metadati presenti in `VG_00050.vg`, e cioè i metadati corretti più recenti archiviati per il gruppo di volumi. L'opzione `restorefile` indica al comando `pvcreate` di rendere il nuovo volume fisico compatibile con quello vecchio presente sul gruppo di volumi, assicurando che la nuova versione dei metadati non venga posizionata dove il volume fisico precedente conteneva i dati (tale comportamento si potrebbe verificare se per esempio, il comando `pvcreate` originale avesse usato gli argomenti della linea di comando per controllare il posizionamento dei metadati, o se il volume fisico fosse stato creato originariamente utilizzando una versione diversa di software con impostazioni predefinite differenti). Il comando `pvcreate` sovrascrive solo le aree dei metadati di LVM e non interessa le aree dei dati esistenti.

```
[root@link-07 backup]# pvcreate --uuid "FmGRh3-zhok-iVI8-7qTD-S5BI-MAEN-NYM5Sk" --restorefile /etc/lvm/archive/VG_00050.vg /dev/sdh1
Physical volume "/dev/sdh1" successfully created
```

Successivamente potrete utilizzare `vgcfgrestore` per ripristinare i metadati del gruppo di volumi.

```
[root@link-07 backup]# vgcfgrestore VG
Restored volume group VG
```

Ora è possibile visualizzare i volumi logici.

```
[root@link-07 backup]# lvs -a -o +devices
  LV      VG      Attr      LSize   Origin Snap%   Move Log Copy%  Devices
  stripe VG      -wi---  300.00G                                     /dev/sdh1
(0),/dev/sda1(0)
  stripe VG      -wi---  300.00G                                     /dev/sdh1
(34728),/dev/sdb1(0)
```

I seguenti comandi attivano i volumi e visualizzano i volumi attivi.

```
[root@link-07 backup]# lvchange -ay /dev/VG/stripe
[root@link-07 backup]# lvs -a -o +devices
  LV      VG      Attr      LSize   Origin Snap%   Move Log Copy%  Devices
  stripe VG      -wi-a-  300.00G                                     /dev/sdh1
(0),/dev/sda1(0)
  stripe VG      -wi-a-  300.00G                                     /dev/sdh1
(34728),/dev/sdb1(0)
```

Se i metadati LVM sul disco necessitano di una quantità minima di spazio uguale a quella sovrascritta, questo comando è in grado di ripristinare il physical volume. Se la quantità di spazio sovrascritto oltrepassa l'area dei metadati, allora i dati presenti sul volume potrebbero essere stati interessati da questo processo. Per recuperare i dati è possibile utilizzare il comando `fsck`.

6.5. SOSTITUZIONE DI UN VOLUME FISICO MANCANTE

Se un volume fisico fallisce o deve essere sostituito sarà possibile etichettare nel gruppo di volumi esistente un nuovo volume fisico seguendo la stessa procedura per il recupero dei metadati del volume fisico descritta nella [Sezione 6.4, «Recupero dei metadati del volume fisico»](#). Usate le opzioni `--partial` e `--verbose` del comando `vgdisplay`, per visualizzare gli UUID e le dimensioni di ogni volume fisico non più esistente. Se desiderate sostituire un altro volume fisico con la stessa dimensione, usate il comando `pvcreate` con le opzioni `--restorefile` e `--uuid` per inizializzare un nuovo dispositivo con lo stesso UUID del volume fisico mancante. Successivamente sarà possibile utilizzare `vgcfgrestore` per recuperare i metadati del gruppo di volumi.

6.6. RIMOZIONE DEI VOLUMI FISICI PERSI DA UN GRUPPO DI VOLUMI

Nell'evento di una perdita di un volume fisico sarà possibile attivare i volumi fisici restanti nel gruppo di volumi tramite l'opzione `--partial` del comando `vgchange`. È possibile rimuovere tutti i volumi logici che utilizzano il volume fisico dal gruppo di volumi tramite l'opzione `--removemissing` del comando `vgreduce`.

È consigliato eseguire il comando `vgreduce` con l'opzione `--test`, per verificare ciò che state eliminando.

Come la maggior parte delle operazioni LVM il comando `vgreduce` è reversibile ciò significa che è possibile utilizzare immediatamente il comando `vgcfgrestore`, per ripristinare i metadati del gruppo di volumi al loro stato precedente. Per esempio, se avete usato l'opzione `--removemissing` del comando `vgreduce` senza l'opzione `--test`, e se avete rimosso i volumi logici che desideravate mantenere, sarà ancora possibile sostituire il volume fisico ed utilizzare un altro comando `vgcfgrestore` per ritornare lo stato del gruppo di volumi al suo stato precedente.

6.7. ESTENSIONI DISPONIBILI INSUFFICIENTI PER UN VOLUME LOGICO

È possibile visualizzare il messaggio "Insufficient free extents" durante la creazione di un volume logico ma credete di avere un numero sufficiente di estensioni in base all'output dei comandi **vgdisplay** o **vgs**. Ciò è causato dai suddetti comandi poichè essi eseguono una approssimazione a due cifre decimali per fornire un output facilmente leggibile da parte dell'utente. Per specificare la dimensione esatta, utilizzate il conteggio dell'estensione fisica disponibile invece dei multipli di byte per determinare la dimensione del volume logico.

Il comando **vgdisplay** per default include questa riga la quale indica le estensioni fisiche disponibili.

```
# vgdisplay
--- Volume group ---
...
Free  PE / Size      8780 / 34.30 GB
```

Alternativamente potrete utilizzare le opzioni **vg_free_count** e **vg_extent_count** di **vgs**, per visualizzare le estensioni disponibili ed il numero totale di estensioni.

```
[root@tng3-1 ~]# vgs -o +vg_free_count,vg_extent_count
VG      #PV #LV #SN Attr   VSize  VFree  Free #Ext
testvg   2   0   0 wz--n- 34.30G 34.30G 8780 8780
```

Con 8780 estensioni fisiche disponibili, è possibile eseguire il seguente comando utilizzando l'opzione **l** minuscola per usare le estensioni invece dei byte:

```
# lvcreate -l8780 -n testlv testvg
```

Verranno utilizzate tutte le estensioni disponibili nel gruppo di volumi.

```
# vgs -o +vg_free_count,vg_extent_count
VG      #PV #LV #SN Attr   VSize  VFree  Free #Ext
testvg   2   1   0 wz--n- 34.30G    0    0 8780
```

Alternativamente potrete estendere il volume logico in modo da usare la percentuale dello spazio rimasto nel gruppo di volumi con l'opzione **-l** del comando **lvcreate**. Per informazioni consultate [Sezione 4.4.1, «Creazione di volumi logici lineari»](#).

CAPITOLO 7. AMMINISTRAZIONE DI LVM CON LA GUI DI LVM

In aggiunta alla Command Line Interface (CLI), LVM fornisce una Graphical User Interface (GUI), la quale può essere utilizzata per configurare i volumi logici LVM. È possibile utilizzare questa utilità digitando `system-config-lvm`. Il capitolo di LVM della *Storage Administration Guide* fornisce le informazioni utili passo-dopo-passo su come configurare un volume logico LVM utilizzando questa utilità.

APPENDICE A. DEVICE MAPPER

Il Device Mapper è un driver del kernel in grado di fornire una struttura generica per la gestione del volume. Fornisce un metodo generico per la creazione di dispositivi mappati, i quali possono essere usati come volumi logici. Esso non è a conoscenza dei gruppi di volumi o dei formati dei metadati.

Il Device Mapper fornisce una base per un numero di tecnologie di livello superiore. In aggiunta a LVM, il Device-Mapper multipath ed il comando `dmraid` utilizzano il Device Mapper. L'interfaccia dell'applicazione per il Device Mapper è la chiamata del sistema `ioctl`. L'interfaccia utente è il comando `dmsetup`.

I volumi logici LVM vengono attivati utilizzando il Device Mapper. Ogni volume logico viene tradotto in un dispositivo mappato. Ogni segmento viene tradotto in una riga nella tabella di mappatura che descrive il dispositivo. Il Device Mapper supporta una varietà di destinazioni di mappatura inclusa una mappatura lineare, segmentata ed una mappatura d'errore. Per esempio, due dischi possono essere concatenati in un volume logico, con un paio di mappature lineari, uno per ogni disco. Quando LVM crea un volume, esso crea un dispositivo device-mapper sottostante il quale può essere interrogato con il comando `dmsetup`. Per informazioni sul formato dei dispositivi in una tabella di mappatura, consultare [Sezione A.1, «Tabella di mappatura del dispositivo»](#). Per informazioni su come utilizzare il comando `dmsetup` per interrogare un dispositivo consultare [Sezione A.2, «Il comando dmsetup»](#).

A.1. TABELLA DI MAPPATURA DEL DISPOSITIVO

Il dispositivo mappato viene definito da una tabella la quale specifica come mappare ogni gamma dei settori logici utilizzando una mappatura supportata della Tabella del dispositivo. La tabella per un dispositivo mappato viene creata da un elenco di righe dal formato:

```
start length mapping [mapping_parameters...]
```

Nella prima riga di una tabella del Device Mapper il parametro *start* deve essere uguale a 0. I parametri *start + length* su di una riga devono essere uguali a *start* sulla riga successiva. I parametri di mappatura da specificare sulla riga della tabella dipendono dal tipo di *mapping* specificato sulla riga.

Le dimensioni nel Device Mapper sono sempre specificate in settori (512 byte).

Quando un dispositivo viene specificato come parametro di mappatura nel Device Mapper, esso può essere indicato con il nome del dispositivo all'interno del filesystem (per esempio `/dev/hda`), o dai numeri minore e maggiore nel formato *major:minor*. Il formato *major:minor* viene preferito poiché evita i lookup del percorso.

Di seguito viene riportato un esempio di tabella di mappatura per un dispositivo. In questa tabella sono presenti quattro destinazioni lineari:

```
0 35258368 linear 8:48 65920
35258368 35258368 linear 8:32 65920
70516736 17694720 linear 8:16 17694976
88211456 17694720 linear 8:16 256
```

Nei primi 2 parametri di ogni riga vengono riportati il blocco d'inizio del segmento e la sua lunghezza. Il parametro successivo è il target di mappatura, il quale in qualsiasi altro caso in questo esempio è *lineare*. Il resto della riga consiste in parametri per un target *lineare*.

Le seguenti sottosezioni descrivono il formato delle seguenti mappature:

- lineare
- segmentato
- mirror
- snapshot e snapshot-origin
- error
- zero
- multipath
- crypt

A.1.1. Target di mappatura lineare

Un target di mappatura lineare mappa una gamma continua di blocchi su un altro dispositivo a blocchi. Il formato di un target lineare è il seguente:

```
start length linear device offset
```

start

blocco iniziale in un dispositivo virtuale

length

lunghezza di questo segmento

device

dispositivo a blocchi, indicato dal nome del dispositivo nel filesystem o dai numeri maggiore e minore nel formato *major:minor*

offset

offset iniziale della mappatura sul dispositivo

Il seguente esempio mostra un target lineare con un blocco iniziale nel dispositivo virtuale di 0, una lunghezza del segmento di 1638400, una coppia di numeri major:minor 8:2, ed un offset iniziale per il dispositivo di 41146992.

```
0 16384000 linear 8:2 41156992
```

Il seguente esempio mostra un target lineare con `/dev/hda` specificato come parametro del dispositivo.

```
0 20971520 linear /dev/hda 384
```

A.1.2. Target per la mappatura segmentata

Il target per la mappatura segmentata supporta il processo di stripping attraverso i dispositivi fisici. Accetta come argomenti il numero di segmenti e la dimensione del segmento, seguiti da un elenco del settore e nome del dispositivo. Il formato di un target segmentato è il seguente:

```
start length striped #stripes chunk_size device1 offset1 ... deviceN offsetN
```

È disponibile un set di parametri *device* e *offset* per ogni striscia.

start

blocco iniziale in un dispositivo virtuale

length

lunghezza di questo segmento

#stripes

numero di segmenti per il dispositivo virtuale

chunk_size

numero dei settori scritti su ogni segmento prima di smistarsi su quello successivo; deve essere una potenza di 2 e grande almeno quanto la dimensione della pagina del kernel

device

dispositivo a blocchi, indicato dal nome del dispositivo nel filesystem o dai numeri maggiore e minore nel formato *major:minor*.

offset

offset iniziale della mappatura sul dispositivo

Il seguente esempio mostra un target segmentato con tre segmenti ed un segmento con una dimensione di 128:

```
0 73728 striped 3 128 8:9 384 8:8 384 8:7 9789824
```

0

blocco iniziale in un dispositivo virtuale

73728

lunghezza di questo segmento

striped 3 128

striscia attraverso tre dispositivi con un segmento di 128 blocchi

8:9

major:minor numeri del primo dispositivo

384

offset d'avvio della mappatura sul primo dispositivo

8:8

numeri major:minor del secondo dispositivo

384

offset d'inizio della mappatura sul secondo dispositivo

8:7

numeri major:minor del terzo dispositivo

9789824

offset d'inizio della mappatura sul terzo dispositivo

Il seguente esempio mostra un target segmentato per due segmenti con dimensioni di 256 KiB, con parametri specificati dai nomi dei dispositivi nel file `system`, e non dai numeri major e minor.

```
0 65536 striped 2 512 /dev/hda 0 /dev/hdb 0
```

A.1.3. Il target di mappatura mirror

Il target di mappatura mirror supporta una mappatura di un dispositivo logico speculare. Il formato di un target speculare è il seguente:

```
start length mirror log_type #logargs logarg1 ... logargN #devs device1
offset1 ... deviceN offsetN
```

start

blocco iniziale in un dispositivo virtuale

length

lunghezza di questo segmento

log_type

I tipi possibili di log ed i rispettivi argomenti sono di seguito riportati:

core

Il mirror è locale ed il mirror log viene conservato nella memoria principale. Questo tipo di log richiede 1 - 3 argomenti:

```
regionsize [[no]sync] [block_on_error]
```

disk

Il mirror è locale ed il mirror log viene conservato sul disco. Questo tipo di log richiede 2 - 4 argomenti:

```
logdevice regionsize [[no]sync] [block_on_error]
```

clustered_core

Il mirror è clusterizzato ed il mirror log viene conservato nella memoria principale. Questo tipo di log richiede 2 - 4 argomenti:

regionsize UUID [[no]sync] [block_on_error]

clustered_disk

Il mirror è clusterizzato ed il mirror log viene conservato sul disco. Questo tipo di log richiede 3 - 5 argomenti:

logdevice regionsize UUID [[no]sync] [block_on_error]

LVM conserva un piccolo log il quale viene utilizzato per controllare le regioni in sincronizzazione con i mirror. L'argomento *regionsize* specifica la dimensione di queste regioni.

In un ambiente clusterizzato l'argomento *UUID* è un identificatore unico associato con il dispositivo mirror log, in modo da poter mantenere lo stato del log sull'intero cluster.

L'argomento *[no]sync* facoltativo può essere utilizzato per specificare il mirror come "in-sync" o "out-of-sync". L'argomento *block_on_error* viene utilizzato per indicare al mirror di rispondere agli errori invece di ignorarli.

#log_args

numero di argomenti per il log specificati nella mappatura.

logargs

gli argomenti di log per il mirror; il numero di argomenti di log forniti viene specificato dal parametro **#log_args**, e gli argomenti di log validi sono determinati dal parametro *log_type*.

#devs

il numero di gambe (o sezioni) nel mirror; viene specificato un dispositivo ed un offset per ogni sezione.

device

dispositivo a blocchi per ogni sezione del mirror, indicato dal nome del dispositivo nel filesystem o dai numeri major e minor nel formato *major:minor*. Un dispositivo a blocchi e l'offset sono specificati per ogni gamba del mirror, come riportato dal parametro **#devs**.

offset

offset d'inizio per la mappatura sul dispositivo. Un dispositivo a blocchi ed un offset sono specificati per ogni sezione del mirror, come indicato dal parametro **#devs**.

Il seguente esempio mostra un target di mappatura del mirror per un mirror clusterizzato con un mirror log sul disco.

```
0 52428800 mirror clustered_disk 4 253:2 1024 UUID block_on_error 3 253:3
0 253:4 0 253:5 0
```

0

blocco iniziale in un dispositivo virtuale

52428800

lunghezza di questo segmento

mirror clustered_disk

target del mirror con un tipo di log il quale specifica un mirror clusterizzato con un mirror log conservato sul disco

4

seguiranno 4 argomenti del mirror log

253:2

major:minor numeri del dispositivo di log

1024

dimensione della regione usata dal mirror log per controllare tutto ciò che è sincronizzato

UUID

UUID del dispositivo del mirror log per conservare le informazioni di un intero cluster

block_on_error

il mirror dovrebbe rispondere agli errori

3

numero di gambe nel mirror

253:3 0 253:4 0 253:5 0

numeri major:minor e offset per dispositivi che costituiscono ogni sezione del mirror

A.1.4. Target di mappatura snapshot e snapshot-origin

Quando si crea la prima snapshot LVM di un volume saranno utilizzati quattro dispositivi Device Mapper:

1. Un dispositivo con una mappatura **lineare** contenente la tabella di mappatura originale del volume sorgente.
2. Un dispositivo con una mappatura **lineare** utilizzato come dispositivo copy-on-write (COW) per il volume sorgente; per ogni scrittura, i dati originali vengono salvati nel dispositivo COW di ogni snapshot per mantenere il proprio contenuto visibile invariato (fino al riempimento del dispositivo COW).
3. Un dispositivo con una mappatura **snapshot** che combina #1 e #2, il quale risulta essere il volume snapshot visibile.
4. Il volume "originale" (il quale utilizza il numero del dispositivo usato dal volume sorgente originale), la cui tabella viene sostituita da una mappatura "snapshot-origin" dal dispositivo #1.

Uno schema fisso per i nomi usato per creare questi dispositivi. Per esempio, è possibile utilizzare i seguenti comandi per creare un volume LVM chiamato **base**, ed un volume snapshot chiamato **snap** basato sul quel volume.

```
# lvcreate -L 1G -n base volumeGroup
# lvcreate -L 100M --snapshot -n snap volumeGroup/base
```

Ciò genera quattro dispositivi i quali verranno visualizzati con i seguenti comandi:

```
# dmsetup table|grep volumeGroup
volumeGroup-base-real: 0 2097152 linear 8:19 384
volumeGroup-snap-cow: 0 204800 linear 8:19 2097536
volumeGroup-snap: 0 2097152 snapshot 254:11 254:12 P 16
volumeGroup-base: 0 2097152 snapshot-origin 254:11

# ls -lL /dev/mapper/volumeGroup-*
brw----- 1 root root 254, 11 29 ago 18:15 /dev/mapper/volumeGroup-base-
real
brw----- 1 root root 254, 12 29 ago 18:15 /dev/mapper/volumeGroup-snap-
cow
brw----- 1 root root 254, 13 29 ago 18:15 /dev/mapper/volumeGroup-snap
brw----- 1 root root 254, 10 29 ago 18:14 /dev/mapper/volumeGroup-base
```

Il formato per il target **snapshot-origin** è il seguente:

```
start length snapshot-origin origin
```

start

blocco iniziale in un dispositivo virtuale

length

lunghezza di questo segmento

origin

volume di base della snapshot

snapshot-origin avrà normalmente uno o più snapshot basati su di esso. I processi di lettura verranno mappati direttamente sul dispositivo di supporto. Per ogni processo di scrittura i dati originali saranno salvati nel dispositivo COW di ogni snapshot in modo da mantenere il proprio contenuto visibile invariato fino all'utilizzo completo del dispositivo COW.

Il formato per il target **snapshot** è il seguente:

```
start length snapshot origin COW-device P|N chunksize
```

start

blocco iniziale in un dispositivo virtuale

length

lunghezza di questo segmento

origin

volume di base della snapshot

COW-device

Dispositivo sul quale vengono conservati sezioni di dati modificati

P|N

P (Persistent) o N (Not persistent); indicano se snapshot sopravviverà dopo il riavvio. Per snapshot transitorie (N), una quantità minore di metadata deve essere salvata sul disco; essi possono essere conservati in memoria dal kernel.

chunksize

Dimensione in settori di segmenti di dati modificati che verranno archiviati sul dispositivo COW.

Il seguente esempio mostra un target **snapshot-origin** con un dispositivo d'origine di 254:11.

```
0 2097152 snapshot-origin 254:11
```

Il seguente esempio mostra un target **snapshot** con un dispositivo d'origine di 254:11, ed un dispositivo COW di 254:12. Il dispositivo snapshot è persistente dopo ogni riavvio e la dimensione del segmento per i dati conservati sul dispositivo COW è di 16 settori.

```
0 2097152 snapshot 254:11 254:12 P 16
```

A.1.5. Target di mappatura 'error'

Con un target di mappatura **error**, qualsiasi operazione I/O per il settore mappato fallirà.

Un target di mappatura **error** può essere utilizzato a scopo di prova. Per provare il comportamento di un dispositivo durante una condizione d'errore, create una mappatura del dispositivo con un settore corrotto nel mezzo del dispositivo stesso, oppure cambiate la sezione di un mirror e sostituirla con un target d'errore.

È possibile utilizzare un target **error** al posto di un dispositivo corrotto, così facendo eviterete sospensioni o vari tentativi sul dispositivo in questione. Esso può servire come target intermedio mentre cercherete di riorganizzare i metadata LVM in presenza di processi falliti.

Il target di mappatura **error** non accetta parametri aggiuntivi oltre ai parametri *start* e *length*.

Il seguente esempio mostra un target **error**.

```
0 65536 error
```

A.1.6. Target di mappatura zero

Il target di mappatura **zero** è un dispositivo a blocchi equivalente a `/dev/zero`. Un processo di lettura per questo tipo di mappatura ritorna blocchi di zero. I dati scritti su questa mappatura vengono scartati, ma il processo di scrittura avrà successo. Il target di mappatura **zero** non accetta parametri aggiuntivi oltre ai parametri *start* e *length*.

Il seguente esempio mostra un target zero per un dispositivo a 16Tb.

```
0 65536 zero
```

A.1.7. Il target di mappatura multipath

Il target di mappatura multipath supporta la mappatura di un dispositivo 'multipathed'. Il formato per il target `multipath` è il seguente:

```
start length multipath #features [feature1 ... featureN] #handlerargs
[handlerarg1 ... handlerargN] #pathgroups pathgroup pathgroupargs1 ...
pathgroupargsN
```

È disponibile un set di parametri *pathgroupargs* per ogni gruppo di percorsi.

start

blocco iniziale in un dispositivo virtuale

length

lunghezza di questo segmento

#features

Il numero di funzioni multipath seguito dalle funzioni in questione. Se questo parametro è zero allora non sarà disponibile alcun parametro *feature*, ed il parametro di mappatura del dispositivo successivo è *#handlerargs*. Attualmente è supportata solo una funzione multipath, **queue_if_no_path**. Ciò indica che il dispositivo 'multipathed' in questione è attualmente impostato per mettere in coda le operazioni I/O se non è disponibile alcun percorso.

Per esempio, se l'opzione **no_path_retry**, nel file `multipath.conf`, è stata impostata in modo da mettere in coda le operazioni I/O solo fino a quando tutti i percorsi sono stati contrassegnati come falliti dopo un certo numero di tentativi fatti per utilizzare il percorso in questione, la mappatura apparirà nel modo seguente fino a quando tutti i controllori del percorso non avranno fallito il numero di controlli specificato.

```
0 71014400 multipath 1 queue_if_no_path 0 2 1 round-robin 0 2 1 66:128 \
1000 65:64 1000 round-robin 0 2 1 8:0 1000 67:192 1000
```

Dopo che tutti i controllori del percorso hanno fallito il numero di controlli specificato la mappatura apparirà nel modo seguente.

```
0 71014400 multipath 0 0 2 1 round-robin 0 2 1 66:128 1000 65:64 1000 \
round-robin 0 2 1 8:0 1000 67:192 1000
```

#handlerargs

Il numero di argomenti del gestore hardware seguito dagli argomenti. Un gestore hardware specifica il modulo che verrà utilizzato per eseguire le azioni hardware specifiche durante lo smistamento tra gruppi di percorsi o durante la gestione degli errori I/O. Se impostato su 0 il parametro successivo è *#pathgroups*.

#pathgroups

Il numero dei gruppi di percorsi. Un gruppo di percorsi rappresenta un insieme di percorsi attraverso i quali un dispositivo 'multipathed' bilancerà il carico. È disponibile un set di parametri *pathgroupargs* per ogni gruppo di percorsi.

pathgroup

Il gruppo di percorsi successivo da provare.

pathgroupsargs

Ogni gruppo di percorsi presenta i seguenti argomenti:

```
pathselector #selectorargs #paths #pathargs device1 ioreqs1 ... deviceN ioreqsN
```

È disponibile un set di argomenti per ogni percorso presente nel gruppo di percorsi.

pathselector

Specifica l'algoritmo utilizzato per determinare il percorso, presente all'interno del gruppo, da utilizzare per l'operazione I/O successiva.

#selectorargs

Il numero di argomenti del selettore del percorso che segue questo argomento nella mappatura multipath. Attualmente il valore di questo argomento è sempre 0.

#paths

Il numero di percorsi in questo gruppo di percorsi.

#pathargs

Il numero di argomenti del percorso specificati per ogni percorso in questo gruppo. Attualmente questo numero è sempre 1, l'argomento *ioreqs*.

device

Il numero del dispositivo a blocchi del percorso, indicato dai numeri major e minor nel formato *major:minor*

ioreqs

Il numero delle richieste I/O per l'instradamento per questo percorso prima di smistarsi sul percorso successivo nel gruppo corrente.

Figura A.1, «Target di mappatura Multipath» mostra il formato di un target multipath con due gruppi di percorsi.

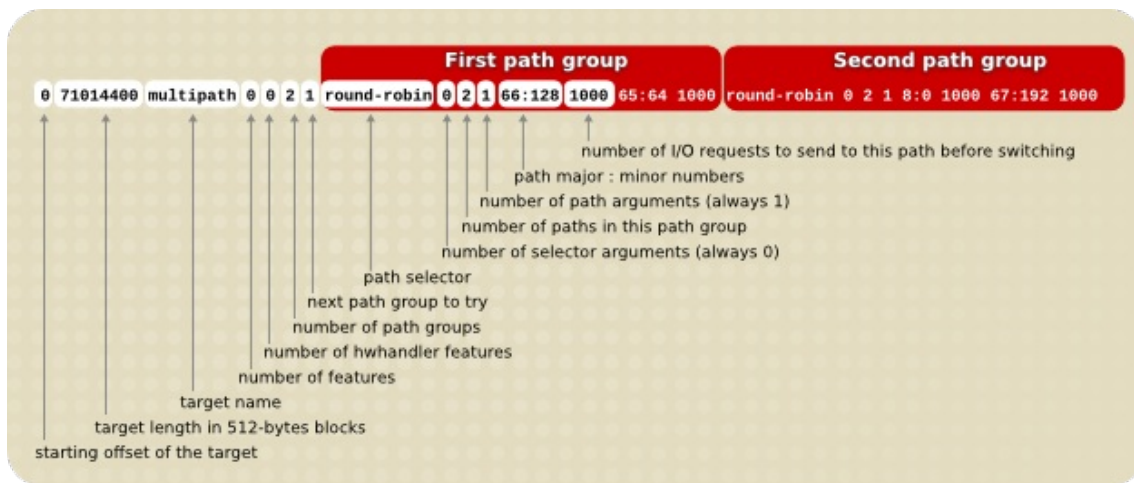


Figura A.1. Target di mappatura Multipath

Il seguente esempio mostra una definizione del target di failover per lo stesso dispositivo multipath. In questo target sono presenti quattro gruppi di percorsi con un solo percorso aperto per gruppo, in questo modo il dispositivo multipathed userà solo un percorso per volta.

```
0 71014400 multipath 0 0 4 1 round-robin 0 1 1 66:112 1000 \
round-robin 0 1 1 67:176 1000 round-robin 0 1 1 68:240 1000 \
round-robin 0 1 1 65:48 1000
```

Il seguente esempio mostra una definizione completa del target (multibus) per lo stesso dispositivo multipathed. In questo target è presente un solo gruppo di percorsi nel quale sono inclusi tutti i percorsi. Con questa impostazione multipath suddivide il carico in modo uniforme su tutti i percorsi.

```
0 71014400 multipath 0 0 1 1 round-robin 0 4 1 66:112 1000 \
67:176 1000 68:240 1000 65:48 1000
```

Per maggiori informazioni sul multipathing consultare la documentazione *Come utilizzare il Device Mapper Multipath*

A.1.8. Target di mappatura crypt

Il target **crypt** cifra i dati che passano attraverso il dispositivo specificato. Esso utilizza il kernel Crypto API.

Il formato per il target **crypt** è il seguente:

```
start length crypt cipher key IV-offset device offset
```

start

blocco iniziale in un dispositivo virtuale

length

lunghezza di questo segmento

cipher

Il Cipher consiste in ***cipher[-chainmode]-ivmode[:iv options]***.

cipher

I Cipher disponibili sono riportati in `/proc/crypto` (per esempio, `aes`).

chainmode

Utilizza sempre `cbc`. Non utilizzare `ebc`; esso non utilizza un initial vector (IV).

ivmode[:iv options]

IV è un initial vector utilizzato per variare la cifratura. La modalità IV può essere `plain` o `essiv:hash`. Un *ivmode* di `-plain` utilizza il numero del settore (più l'offset IV) di IV. Un *ivmode* di `-essiv` è il modo migliore per non avere problemi con il watermark.

key

Chiave di cifratura fornita in esadecimale

IV-offset

Offset Initial Vector (IV)

device

dispositivo a blocchi, indicato dal nome del dispositivo nel filesystem o dai numeri maggiore e minore nel formato *major:minor*

offset

offset iniziale della mappatura sul dispositivo

Il seguente è un esempio di un target `crypt`.

```
0 2097152 crypt aes-plain 0123456789abcdef0123456789abcdef 0 /dev/hda 0
```

A.2. IL COMANDO DMSETUP

Il comando `dmsetup` è un wrapper della linea di comando per le comunicazioni con il Device Mapper. Per informazioni generali del sistema sui dispositivi LVM, le opzioni `info`, `ls`, `status`, e `deps` del comando `dmsetup` potranno risultare utili, come descritto nelle seguenti sottosezioni.

Per informazioni sulle opzioni aggiuntive e capacità di `dmsetup` consultate la pagina `man (8) dmsetup`.

A.2.1. Il comando `dmsetup info`

Il comando `dmsetup info device` fornisce le informazioni sui dispositivi del Device Mapper. Se non specificate alcun nome l'output conterrà le informazioni di tutti i dispositivi del Device Mapper attualmente configurati. Se al contrario viene specificato un dispositivo, le informazioni fornite riguarderanno solo il dispositivo interessato.

Il comando `dmsetup info` fornisce le informazioni nelle seguenti categorie:

Nome

Il nome del dispositivo. Un dispositivo LVM viene indicato come nome del gruppo di volumi e nome del volume logico separato da un trattino. Un trattino nel nome originale viene tradotto in due trattini.

Stato

Gli stati possibili del dispositivo sono **SUSPENDED**, **ACTIVE**, e **READ-ONLY**. Il comando **dmsetup suspend** imposta lo stato su **SUSPENDED**. Quando un dispositivo risulta sospeso, tutte le operazioni I/O per quel dispositivo verranno arrestate. Il comando **dmsetup resume** ripristina lo stato del dispositivo su **ACTIVE**.

Read Ahead

Il numero di blocchi dati letti a priori dal sistema per qualsiasi file aperto sul quale sono in corso le operazioni di lettura. Nell'impostazione predefinita il kernel seleziona automaticamente un valore idoneo. È possibile modificare questo valore con l'opzione **- - readahead** del comando **dmsetup**.

Tabelle presenti

Gli stati possibili per questa categoria sono **LIVE** e **INACTIVE**. Uno stato **INACTIVE** indica che una tabella è stata caricata e verrà intercambiata quando il comando **dmsetup resume** ripristina lo stato di un dispositivo su **ACTIVE**, a questo punto lo stato della tabella diventerà **LIVE**. Per maggiori informazioni consultare la pagina man di **dmsetup**.

Conteggio apertura

L'open reference count indica il numero di volte che un dispositivo viene aperto. Un comando **mount** apre il dispositivo.

Numero eventi

Il numero corrente degli eventi. L'emissione del comando **dmsetup wait n** permette all'utente di aspettare che l'evento cambi, bloccando la chiamata fino a quando viene ricevuta.

Maggiore, minore

I numeri maggiore e minore del dispositivo

Numero di target

Il numero di frammenti che costituisce un dispositivo. Per esempio, un dispositivo lineare che si estende su 3 dischi avrà 3 target. Un dispositivo lineare composto dall'inizio e fine di un disco, ma non dal centro, avrà 2 target.

UUID

UUID del dispositivo.

Il seguente esempio mostra un output parziale per il comando **dmsetup info**.

```
[root@ask-07 ~]# dmsetup info
Name:          testgfsvg-testgfs1v1
State:         ACTIVE
Read Ahead:    256
Tables present: LIVE
Open count:    0
Event number:  0
```

```

Major, minor:      253, 2
Number of targets: 2
UUID: LVM-K528WUGQgPadNXYcFrrf9LnPlUMswgkCkpgPIgYzSvigM7SfewCypddNSwtNzc2N
...
Name:              VolGroup00-LogVol00
State:             ACTIVE
Read Ahead:       256
Tables present:   LIVE
Open count:       1
Event number:     0
Major, minor:     253, 0
Number of targets: 1
UUID: LVM-t0cS1kqFV9drb0X1Vr8sxeYP0tqcrpdegyqj5lZxe45JMGlmvtqLmbLpBcenh2L3

```

A.2.2. Il comando `dmsetup ls`

È possibile elencare i nomi dei dispositivi mappati con il comando `dmsetup ls` altresì sarà possibile elencare i dispositivi con almeno un target di tipo specifico con il comando `dmsetup ls --target target_type`. Per altre opzioni di `dmsetup ls`, consultare la pagina man `dmsetup`.

Il seguente esempio mostra il comando usato per elencare i nomi dei dispositivi mappati attualmente configurati.

```

[root@ask-07 ~]# dmsetup ls
testgfsvg-testgfslv3 (253, 4)
testgfsvg-testgfslv2 (253, 3)
testgfsvg-testgfslv1 (253, 2)
VolGroup00-LogVol01 (253, 1)
VolGroup00-LogVol00 (253, 0)

```

Il seguente esempio mostra il comando usato per elencare i nomi delle mappature dei mirror attualmente configurati.

```

[root@grant-01 ~]# dmsetup ls --target mirror
lock_stress-grant--02.1722 (253, 34)
lock_stress-grant--01.1720 (253, 18)
lock_stress-grant--03.1718 (253, 52)
lock_stress-grant--02.1716 (253, 40)
lock_stress-grant--03.1713 (253, 47)
lock_stress-grant--02.1709 (253, 23)
lock_stress-grant--01.1707 (253, 8)
lock_stress-grant--01.1724 (253, 14)
lock_stress-grant--03.1711 (253, 27)

```

Le configurazioni LVM presenti su dispositivi device mapper o con percorsi multipli possono essere complessi da classificare. Il comando `dmsetup ls` fornisce una opzione `--tree` la quale mostra le dipendenze presenti tra i dispositivi sotto forma di albero, come riportato nel seguente esempio:

```

# dmsetup ls --tree
vgtest-lvmir (253:13)
├─vgtest-lvmir_mimage_1 (253:12)
│   └─mpathep1 (253:8)
│       └─mpathe (253:5)
│           └─ (8:112)

```

```

└─ (8:64)
└─vgtest-lvmir_mimage_0 (253:11)
    └─mpathcp1 (253:3)
        └─mpathc (253:2)
            └─ (8:32)
                └─ (8:16)
└─vgtest-lvmir_mlog (253:4)
    └─mpathfp1 (253:10)
        └─mpathf (253:6)
            └─ (8:128)
                └─ (8:80)

```

A.2.3. Il comando `dmsetup status`

Il comando `dmsetup status device` fornisce le informazioni sullo stato per ogni target in un dispositivo specifico. Se non specificate alcun nome, l'output conterrà le informazioni di tutti i dispositivi Device Mapper attualmente configurati. È possibile elencare lo stato dei dispositivi con almeno un target di tipo specifico specificato con il comando `dmsetup status --target target_type`.

Il seguente esempio mostra il comando usato per elencare lo stato dei target in tutti i dispositivi mappati attualmente configurati.

```

[root@ask-07 ~]# dmsetup status
testgfsvg-testgfs1v3: 0 312352768 linear
testgfsvg-testgfs1v2: 0 312352768 linear
testgfsvg-testgfs1v1: 0 312352768 linear
testgfsvg-testgfs1v1: 312352768 50331648 linear
VolGroup00-LogVol01: 0 4063232 linear
VolGroup00-LogVol00: 0 151912448 linear

```

A.2.4. Il comando `dmsetup deps`

Il comando `dmsetup deps device` fornisce un elenco di coppie (maggiore, minore) per dispositivi indicati dalla tabella di mappatura per il dispositivo specificato. Se non specificate il nome del dispositivo, le informazioni contenute dall'output riguarderanno tutti i dispositivi Device Mapper attualmente configurati.

Il seguente esempio mostra il comando usato per elencare le dipendenze di tutti i dispositivi mappati attualmente configurati.

```

[root@ask-07 ~]# dmsetup deps
testgfsvg-testgfs1v3: 1 dependencies : (8, 16)
testgfsvg-testgfs1v2: 1 dependencies : (8, 16)
testgfsvg-testgfs1v1: 1 dependencies : (8, 16)
VolGroup00-LogVol01: 1 dependencies : (8, 2)
VolGroup00-LogVol00: 1 dependencies : (8, 2)

```

Il seguente esempio mostra il comando usato per elencare le dipendenze solo del dispositivo `lock_stress-grant--02.1722`:

```

[root@grant-01 ~]# dmsetup deps lock_stress-grant--02.1722
3 dependencies : (253, 33) (253, 32) (253, 31)

```

A.3. SUPPORTO DEVICE MAPPER PER IL DEVICE MANAGER UDEV

Il ruolo primario del device manager **udev** è quello di fornire un metodo dinamico per l'impostazione dei nodi nella directory `/dev`. La creazione di questi nodi viene eseguita tramite l'implementazione delle regole **udev** nello spazio utente. Le suddette regole sono processate sugli eventi **udev** inviati dal kernel come risultato di una aggiunta, rimozione o modifica di dispositivi particolari. Ciò fornisce un meccanismo centrale conveniente per un supporto di tipo hotplug.

Oltre a creare i nodi il device manager **udev** è in grado di creare qualsiasi link simbolico con i rispettivi nomi, fornendo agli utenti la possibilità di scegliere i nomi e la struttura della directory personalizzati nella directory `/dev`.

Ogni evento **udev** contiene le informazioni di base sul dispositivo processato, ad esempio il nome, il sottosistema a cui appartiene ed il tipo di dispositivo, i numeri maggiore e minore usati ed il tipo di evento. Con questi dati e con la possibilità di accedere a tutte le informazioni disponibili nella directory `/sys` accessibile anche all'interno delle regole **udev**, gli utenti saranno in grado di utilizzare filtri semplici ed eseguire le regole in base alle suddette informazioni.

Il device manager **udev** fornisce anche un metodo centralizzato per l'impostazione dei permessi dei nodi. Un utente sarà in grado di aggiungere un set di regole personalizzate per definire i permessi per qualsiasi dispositivo specificato da qualsiasi informazione disponibile durante la processazione dell'evento.

È possibile aggiungere direttamente gli hook di programma nelle regole **udev**. Il device manager **udev** è in grado di chiamare i suddetti programmi in modo da fornire una ulteriore processazione per la gestione degli eventi. Inoltre il programma è in grado di esportare le variabili dell'ambiente come risultato di questo processo. Ogni risultato dato può essere usato nelle regole come risorsa supplementare.

Qualsiasi software che utilizza la libreria **udev** è in grado di ricevere e processare gli eventi **udev** con tutte le informazioni disponibili, in questo modo la processazione non dipende solo dal demone **udev**.

A.3.1. Integrazione udev con il Device Mapper

Con RHEL 6 il Device Mapper fornisce un supporto diretto all'integrazione **udev**. Tale operazione sincronizza il Device Mapper con tutta la processazione di **udev** relativa ai dispositivi del Device Mapper, incluso i dispositivi LVM. La sincronizzazione è necessaria poichè l'applicazione delle regole nel demone di **udev** è una forma di processazione parallela al programma d'origine delle modifiche del dispositivo (come ad esempio `dmsetup` e LVM). Senza questo supporto il tentativo di rimozione di un dispositivo ancora aperto e processato dalle regole **udev** a causa di alcune modifiche di eventi precedenti risultava essere un problema comune; questa tendenza era molto comune se il tempo intercorso tra le modifiche era breve.

La release RHEL 6 fornisce le regole **udev** ufficialmente supportate per i dispositivi del Device Mapper in generale e LVM. [Tabella A.1, «Regole udev per dispositivi Device-Mapper»](#) riassume le suddette regole presenti in `/lib/udev/rules.d`.

Tabella A.1. Regole udev per dispositivi Device-Mapper

Nome file	Descrizione
-----------	-------------

Nome file	Descrizione
10-dm.rules	<p>Contiene le regole del Device Mapper di base/generali e crea i link simbolici in /dev/mapper con un target /dev/dm-<i>N</i> dove <i>N</i> è /dev/dm-<i>N</i> un numero assegnato dinamicamente ad un dispositivo dal kernel (/dev/dm-<i>N</i> è un nodo)</p> <p>NOTA BENE: I nodi /dev/dm-<i>N</i> non devono <i>mai</i> essere usati negli script per accedere al dispositivo poichè il numero <i>N</i> è assegnato dinamicamente e modificato con la sequenza usata per attivare i dispositivi. Per questo motivo usare i nomi veri presenti nella directory /dev/mapper. Questo layout è usato per supportare i requisiti di udev su come creare i nodi/link simbolici.</p>
11-dm-lvm.rules	<p>Contiene le regole applicate ai dispositivi LVM e crea i link simbolici per i volumi logici del gruppo di volumi. I link simbolici vengono creati nella directory /dev/vgname con un target /dev/dm-<i>N</i>.</p> <p>NOTA BENE: Per essere conformi con gli standard sui nomi delle regole future per i sistemi secondari del Device Mapper, le regole udev dovranno seguire il formato 11-dm-<i>subsystem_name</i>.rules. Qualsiasi utente libdevmapper che fornisce le regole udev dovrà seguire il suddetto standard.</p>
13-dm-disk.rules	<p>Contiene le regole da applicare per tutti i dispositivi del Device Mapper in generale e crea i link simbolici nelle directory /dev/disk/by-id, /dev/disk/by-uuid e /dev/disk/by-uuid</p>
95-dm-notify.rules	<p>Contiene le regole per la notifica dei processi in attesa che utilizzano libdevmapper (proprio come LVM edmssetup). La notifica viene eseguita dopo aver applicato le precedenti regole in modo da assicurare il completamento di qualsiasi processazione da parte di udev. Il processo notificato verrà successivamente riattivato.</p>

È possibile aggiungere regole aggiuntive sui permessi per mezzo del file **12-dm-permissions.rules**. Questo file *non* è installato nella directory **/lib/udev/rules** ma nella directory **/usr/share/doc/device-mapper-*version***. Il file **12-dm-permissions.rules** è un modello il quale contiene suggerimenti su come impostare i permessi, in base ad alcune regole d'esempio; il file contiene gli esempi per alcune situazioni tipo. Sarà possibile modificare questo file e posizionarlo manualmente nella directory **/etc/udev/rules.d** dove potrà preservare le impostazioni.

Queste regole impostano tutte le variabili di base che possono essere usate da qualsiasi altra regola durante la processazione degli eventi.

Le seguenti variabili sono impostate all'interno di `10-dm.rules`:

- **DM_NAME**: Nome dispositivo del Device Mapper
- **DM_UUID**: UUID del dispositivo del Device Mapper
- **DM_SUSPENDED**: lo stato sospeso del dispositivo del Device Mapper
- **DM_UDEV_RULES_VSN**: versione regole `udev` (principalmente per tutte le altre regole per controllare che le variabili precedentemente indicate siano state impostate direttamente da regole ufficiali del Device Mapper)

Le seguenti variabili sono impostate in `11-dm-lvm.rules`:

- **DM_LV_NAME**: nome volume logico
- **DM_VG_NAME**: nome gruppo di volumi
- **DM_LV_LAYER**: nome livello LVM

Tutte queste variabili possono essere usate nel file `12-dm-permissions.rules` per definire un permesso di dispositivi Device Mapper specifici come documentato nel file `12-dm-permissions.rules`.

A.3.2. Comandi ed interfacce che supportano udev

Tabella A.2, «Comandi `dmsetup` per il supporto di `udev`» riassume i comandi `dmsetup` che supportano l'integrazione `udev`.

Tabella A.2. Comandi `dmsetup` per il supporto di `udev`

Comando	Descrizione
<code>dmsetup udevcomplete</code>	Usato per notificare che <code>udev</code> ha terminato la processazione delle regole con il relativo sblocco del processo in attesa (chiamato dalle regole <code>udev</code> in <code>95-dm-notify.rules</code>)
<code>dmsetup udevcomplete_all</code>	Usato per il debugging per sbloccare manualmente tutti i processi in attesa
<code>dmsetup udevcookies</code>	Usato per il debugging per mostrare tutti i cookie esistenti (semafori dell'intero sistema)
<code>dmsetup udevcreatecookie</code>	Usato per la creazione manuale di un cookie (semaforo). Utile per eseguire più processi con una risorsa di sincronizzazione.
<code>dmsetup udevreleasecookie</code>	Usato per attendere la processazione <code>udev</code> relativa a tutti i processi presenti in un cookie di sincronizzazione.

Le opzioni `dmsetup` che supportano l'integrazione `udev` sono di seguito riportate.

--udevcookie

Da definire per tutti i processi **dmsetup** che si desiderano aggiungere in una transazione **udev**. Usato insieme a **udevcreatecookie** e **udevreleasecookie**:

```
COOKIE=$(dmsetup udevcreatecookie)
dmsetup command --udevcookie $COOKIE ....
dmsetup command --udevcookie $COOKIE ....
....
dmsetup command --udevcookie $COOKIE ....
dmsetup udevreleasecookie --udevcookie $COOKIE
```

Invece di usare l'opzione **--udevcookie** sarà possibile esportare la variabile in un ambiente del processo.

```
export DM_UDEV_COOKIE=$(dmsetup udevcreatecookie)
dmsetup command ...
dmsetup command ...
...
dmsetup command ...
```

--noudevrules

Disabilita le regole **udev**. I nodi/link simbolici verranno creati da **libdevmapper** (seguendo la vecchia procedura). Questa opzione viene usata per il debugging se **udev** non funziona correttamente.

--noudevsync

Disabilita la sincronizzazione **udev**. Anche a scopo di debugging.

Per informazioni su **dmsetup** e relative opzioni consultate la pagina man (8) di **dmsetup**.

I comandi LVM supportano le seguenti opzioni le quali supportano l'integrazione **udev**:

- **--noudevrules**: per il comando **dmsetup**, disabilita le regole **udev**.
- **--noudevsync**: per il comando **dmsetup**, disabilita la sincronizzazione **udev**.

Il file **lvm.conf** include le seguenti opzioni in grado di supportare l'integrazione **udev**:

- **udev_rules**: abilita/disabilita globalmente **udev_rules** per tutti i comandi LVM2
- **udev_sync**: abilita/disabilita la sincronizzazione **udev** globalmente per tutti i comandi LVM

Per maggiori informazioni sulle opzioni del file **lvm.conf** consultare i commenti presenti nel file **lvm.conf**.

APPENDICE B. FILE DI CONFIGURAZIONE LVM

LVM supporta file di configurazione multipli. All'avvio del sistema il file di configurazione `lvm.conf` viene caricato dalla directory specificata per mezzo della variabile dell'ambiente `LVM_SYSTEM_DIR`, la quale è impostata per default su `/etc/lvm`.

Il file `lvm.conf` è in grado di specificare i file aggiuntivi di configurazione da caricare. Le impostazioni dei file più recenti sovrascrivono le impostazioni dei file meno recenti. Per visualizzare le impostazioni in uso dopo aver caricato tutti i file di configurazione, eseguite il comando `lvm dumpconfig`.

Per informazioni su come caricare i file aggiuntivi di configurazione, consultate [Sezione C.2, «Tag dell'host»](#).

B.1. FILE DI CONFIGURAZIONE LVM

I seguenti file vengono usati per la configurazione di LVM:

`/etc/lvm/lvm.conf`

file di configurazione centrale letto dai tool.

`etc/lvm/lvm_hosttag.conf`

Per ogni tag dell'host, verrà letto un file di configurazione aggiuntivo se esistente: `lvm_hosttag.conf`. Se il file in questione definisce nuovi tag, allora verranno aggiunti alla lista alcuni file di configurazione supplementari da leggere. Per informazioni sui tag degli host consultate [Sezione C.2, «Tag dell'host»](#).

In aggiunta ai file di configurazione di LVM, il sistema che esegue LVM include i seguenti file i quali interessano l'impostazione del sistema LVM:

`/etc/lvm/.cache`

file di cache del filtro del nome del dispositivo (configurabile).

`/etc/lvm/backup/`

directory per backup automatici dei metadati del gruppo di volumi (configurabili).

`/etc/lvm/archive/`

directory per archivi automatici di metadati per il gruppo di volumi (configurabile in relazione al percorso della directory ed alla cronologia).

`/var/lock/lvm`

In una configurazione host singolo eseguite il lock dei file per prevenire la corruzione dei metadati da parte del parallel tool; in un cluster viene usato il cluster-wide DLM.

B.2. ESEMPIO DI FILE LVM.CONF

Il seguente è un esempio del file di configurazione `lvm.conf`. Il vostro file di configurazione potrebbe essere leggermente diverso.

```
# This is an example configuration file for the LVM2 system.
```

```

# It contains the default settings that would be used if there was no
# /etc/lvm/lvm.conf file.
#
# Refer to 'man lvm.conf' for further information including the file
# layout.
#
# To put this file in a different directory and override /etc/lvm set
# the environment variable LVM_SYSTEM_DIR before running the tools.

# This section allows you to configure which block devices should
# be used by the LVM system.
devices {

    # Where do you want your volume groups to appear ?
    dir = "/dev"

    # An array of directories that contain the device nodes you wish
    # to use with LVM2.
    scan = [ "/dev" ]

    # If several entries in the scanned directories correspond to the
    # same block device and the tools need to display a name for device,
    # all the pathnames are matched against each item in the following
    # list of regular expressions in turn and the first match is used.
    # preferred_names = [ ]

    # Try to avoid using un-descriptive /dev/dm-N names, if present.
    preferred_names = [ "^/dev/mpath/", "^/dev/mapper/mpath",
    "^/dev/[hs]d" ]

    # A filter that tells LVM2 to only use a restricted set of devices.
    # The filter consists of an array of regular expressions. These
    # expressions can be delimited by a character of your choice, and
    # prefixed with either an 'a' (for accept) or 'r' (for reject).
    # The first expression found to match a device name determines if
    # the device will be accepted or rejected (ignored). Devices that
    # don't match any patterns are accepted.

    # Be careful if there are symbolic links or multiple filesystem
    # entries for the same device as each name is checked separately
    against
    # the list of patterns. The effect is that if any name matches any
    'a'
    # pattern, the device is accepted; otherwise if any name matches any
    'r'
    # pattern it is rejected; otherwise it is accepted.

    # Don't have more than one filter line active at once: only one gets
    used.

    # Run vgscan after you change this parameter to ensure that
    # the cache file gets regenerated (see below).
    # If it doesn't do what you expect, check the output of 'vgscan -
    vvvv'.

```

```

# By default we accept every block device:
filter = [ "a/*/" ]

# Exclude the cdrom drive
# filter = [ "r|/dev/cdrom|" ]

# When testing I like to work with just loopback devices:
# filter = [ "a/loop/", "r/*/" ]

# Or maybe all loops and ide drives except hdc:
# filter =[ "a|loop|", "r|/dev/hdc|", "a|/dev/ide|", "r|.*|" ]

# Use anchors if you want to be really specific
# filter = [ "a|^/dev/hda8$|", "r/*/" ]

# The results of the filtering are cached on disk to avoid
# rescanning dud devices (which can take a very long time).
# By default this cache is stored in the /etc/lvm/cache directory
# in a file called '.cache'.
# It is safe to delete the contents: the tools regenerate it.
# (The old setting 'cache' is still respected if neither of
# these new ones is present.)
cache_dir = "/etc/lvm/cache"
cache_file_prefix = ""

# You can turn off writing this cache file by setting this to 0.
write_cache_state = 1

# Advanced settings.

# List of pairs of additional acceptable block device types found
# in /proc/devices with maximum (non-zero) number of partitions.
# types = [ "fd", 16 ]

# If sysfs is mounted (2.6 kernels) restrict device scanning to
# the block devices it believes are valid.
# 1 enables; 0 disables.
sysfs_scan = 1

# By default, LVM2 will ignore devices used as components of
# software RAID (md) devices by looking for md superblocks.
# 1 enables; 0 disables.
md_component_detection = 1

# By default, if a PV is placed directly upon an md device, LVM2
# will align its data blocks with the md device's stripe-width.
# 1 enables; 0 disables.
md_chunk_alignment = 1

# Default alignment of the start of a data area in MB. If set to 0,
# a value of 64KB will be used. Set to 1 for 1MiB, 2 for 2MiB, etc.
# default_data_alignment = 1

# By default, the start of a PV's data area will be a multiple of
# the 'minimum_io_size' or 'optimal_io_size' exposed in sysfs.

```

```

# - minimum_io_size - the smallest request the device can perform
#   w/o incurring a read-modify-write penalty (e.g. MD's chunk size)
# - optimal_io_size - the device's preferred unit of receiving I/O
#   (e.g. MD's stripe width)
# minimum_io_size is used if optimal_io_size is undefined (0).
# If md_chunk_alignment is enabled, that detects the optimal_io_size.
# This setting takes precedence over md_chunk_alignment.
# 1 enables; 0 disables.
data_alignment_detection = 1

# Alignment (in KB) of start of data area when creating a new PV.
# md_chunk_alignment and data_alignment_detection are disabled if set.
# Set to 0 for the default alignment (see: data_alignment_default)
# or page size, if larger.
data_alignment = 0

# By default, the start of the PV's aligned data area will be shifted
by
# the 'alignment_offset' exposed in sysfs. This offset is often 0 but
# may be non-zero; e.g.: certain 4KB sector drives that compensate for
# windows partitioning will have an alignment_offset of 3584 bytes
# (sector 7 is the lowest aligned logical block, the 4KB sectors start
# at LBA -1, and consequently sector 63 is aligned on a 4KB boundary).
# But note that pvcreate --dataalignmentoffset will skip this
detection.
# 1 enables; 0 disables.
data_alignment_offset_detection = 1

# If, while scanning the system for PVs, LVM2 encounters a device-
mapper
# device that has its I/O suspended, it waits for it to become
accessible.
# Set this to 1 to skip such devices. This should only be needed
# in recovery situations.
ignore_suspended_devices = 0

# During each LVM operation errors received from each device are
counted.
# If the counter of a particular device exceeds the limit set here, no
# further I/O is sent to that device for the remainder of the
respective
# operation. Setting the parameter to 0 disables the counters
altogether.
disable_after_error_count = 0

# Allow use of pvcreate --uuid without requiring --restorefile.
require_restorefile_with_uuid = 1
}

# This section allows you to configure the way in which LVM selects
# free space for its Logical Volumes.
#allocation {
#   When searching for free space to extend an LV, the "cling"
#   allocation policy will choose space on the same PVs as the last
#   segment of the existing LV. If there is insufficient space and a
#   list of tags is defined here, it will check whether any of them are

```

```

#   attached to the PVs concerned and then seek to match those PV tags
#   between existing extents and new extents.
#   Use the special tag "@" as a wildcard to match any PV tag.
#
#   Example: LVs are mirrored between two sites within a single VG.
#   PVs are tagged with either @site1 or @site2 to indicate where
#   they are situated.
#
#   cling_tag_list = [ "@site1", "@site2" ]
#   cling_tag_list = [ "@" ]
#}

# This section that allows you to configure the nature of the
# information that LVM2 reports.
log {

    # Controls the messages sent to stdout or stderr.
    # There are three levels of verbosity, 3 being the most verbose.
    verbose = 0

    # Should we send log messages through syslog?
    # 1 is yes; 0 is no.
    syslog = 1

    # Should we log error and debug messages to a file?
    # By default there is no log file.
    #file = "/var/log/lvm2.log"

    # Should we overwrite the log file each time the program is run?
    # By default we append.
    overwrite = 0

    # What level of log messages should we send to the log file and/or
syslog?
    # There are 6 syslog-like log levels currently in use - 2 to 7
inclusive.
    # 7 is the most verbose (LOG_DEBUG).
    level = 0

    # Format of output messages
    # Whether or not (1 or 0) to indent messages according to their
severity
    indent = 1

    # Whether or not (1 or 0) to display the command name on each line
output
    command_names = 0

    # A prefix to use before the message text (but after the command name,
# if selected). Default is two spaces, so you can see/grep the
severity
# of each message.
    prefix = "  "

    # To make the messages look similar to the original LVM tools use:
    #   indent = 0

```

```
#   command_names = 1
#   prefix = " -- "

# Set this if you want log messages during activation.
# Don't use this in low memory situations (can deadlock).
# activation = 0
}

# Configuration of metadata backups and archiving.  In LVM2 when we
# talk about a 'backup' we mean making a copy of the metadata for the
# *current* system.  The 'archive' contains old metadata configurations.
# Backups are stored in a human readable text format.
backup {

    # Should we maintain a backup of the current metadata configuration ?
    # Use 1 for Yes; 0 for No.
    # Think very hard before turning this off!
    backup = 1

    # Where shall we keep it ?
    # Remember to back up this directory regularly!
    backup_dir = "/etc/lvm/backup"

    # Should we maintain an archive of old metadata configurations.
    # Use 1 for Yes; 0 for No.
    # On by default.  Think very hard before turning this off.
    archive = 1

    # Where should archived files go ?
    # Remember to back up this directory regularly!
    archive_dir = "/etc/lvm/archive"

    # What is the minimum number of archive files you wish to keep ?
    retain_min = 10

    # What is the minimum time you wish to keep an archive file for ?
    retain_days = 30
}

# Settings for the running LVM2 in shell (readline) mode.
shell {

    # Number of lines of history to store in ~/.lvm_history
    history_size = 100
}

# Miscellaneous global LVM2 settings
global {

    # The file creation mask for any files and directories created.
    # Interpreted as octal if the first digit is zero.
    umask = 077

    # Allow other users to read the files
    #umask = 022
```



```

# Enabling test mode means that no changes to the on disk metadata
# will be made. Equivalent to having the -t option on every
# command. Defaults to off.
test = 0

# Default value for --units argument
units = "h"

# Since version 2.02.54, the tools distinguish between powers of
# 1024 bytes (e.g. KiB, MiB, GiB) and powers of 1000 bytes (e.g.
# KB, MB, GB).
# If you have scripts that depend on the old behaviour, set this to 0
# temporarily until you update them.
si_unit_consistency = 1

# Whether or not to communicate with the kernel device-mapper.
# Set to 0 if you want to use the tools to manipulate LVM metadata
# without activating any logical volumes.
# If the device-mapper kernel driver is not present in your kernel
# setting this to 0 should suppress the error messages.
activation = 1

# If we can't communicate with device-mapper, should we try running
# the LVM1 tools?
# This option only applies to 2.4 kernels and is provided to help you
# switch between device-mapper kernels and LVM1 kernels.
# The LVM1 tools need to be installed with .lvm1 suffices
# e.g. vgscan.lvm1 and they will stop working after you start using
# the new lvm2 on-disk metadata format.
# The default value is set when the tools are built.
# fallback_to_lvm1 = 0

# The default metadata format that commands should use - "lvm1" or
"lvm2".
# The command line override is -M1 or -M2.
# Defaults to "lvm2".
# format = "lvm2"

# Location of proc filesystem
proc = "/proc"

# Type of locking to use. Defaults to local file-based locking (1).
# Turn locking off by setting to 0 (dangerous: risks metadata
corruption
# if LVM2 commands get run concurrently).
# Type 2 uses the external shared library locking_library.
# Type 3 uses built-in clustered locking.
# Type 4 uses read-only locking which forbids any operations that
might
# change metadata.
locking_type = 1

# Set to 0 to fail when a lock request cannot be satisfied
immediately.
wait_for_locks = 1

```

```

# If using external locking (type 2) and initialisation fails,
# with this set to 1 an attempt will be made to use the built-in
# clustered locking.
# If you are using a customised locking_library you should set this to
0.
    fallback_to_clustered_locking = 1

# If an attempt to initialise type 2 or type 3 locking failed, perhaps
# because cluster components such as clvmd are not running, with this
set
# to 1 an attempt will be made to use local file-based locking (type
1).
    # If this succeeds, only commands against local volume groups will
    proceed.
    # Volume Groups marked as clustered will be ignored.
    fallback_to_local_locking = 1

# Local non-LV directory that holds file-based locks while commands
are
# in progress. A directory like /tmp that may get wiped on reboot is
OK.
    locking_dir = "/var/lock/lvm"

# Whenever there are competing read-only and read-write access
requests for
# a volume group's metadata, instead of always granting the read-only
# requests immediately, delay them to allow the read-write requests to
be
# serviced. Without this setting, write access may be stalled by a
high
# volume of read-only requests.
# NB. This option only affects locking_type = 1 viz. local file-based
# locking.
    prioritise_write_locks = 1

# Other entries can go here to allow you to load shared libraries
# e.g. if support for LVM1 metadata was compiled as a shared library
use
    # format_libraries = "liblvm2format1.so"
    # Full pathnames can be given.

# Search this directory first for shared libraries.
#    library_dir = "/lib"

# The external locking library to load if locking_type is set to 2.
#    locking_library = "liblvm2clusterlock.so"

# Treat any internal errors as fatal errors, aborting the process that
# encountered the internal error. Please only enable for debugging.
    abort_on_internal_errors = 0

# If set to 1, no operations that change on-disk metadata will be
permitted.
# Additionally, read-only commands that encounter metadata in need of
repair

```

```

# will still be allowed to proceed exactly as if the repair had been
# performed (except for the unchanged vg_seqno).
# Inappropriate use could mess up your system, so seek advice first!
metadata_read_only = 0
}

activation {
    # Set to 0 to disable udev synchronisation (if compiled into the
    binaries).
    # Processes will not wait for notification from udev.
    # They will continue irrespective of any possible udev processing
    # in the background. You should only use this if udev is not running
    # or has rules that ignore the devices LVM2 creates.
    # The command line argument --nodevsysnc takes precedence over this
    setting.
    # If set to 1 when udev is not running, and there are LVM2 processes
    # waiting for udev, run 'dmsetup udevcomplete_all' manually to wake
    them up.
    udev_sync = 1

    # Set to 0 to disable the udev rules installed by LVM2 (if built with
    # --enable-udev_rules). LVM2 will then manage the /dev nodes and
    symlinks
    # for active logical volumes directly itself.
    # N.B. Manual intervention may be required if this setting is changed
    # while any logical volumes are active.
    udev_rules = 1

    # How to fill in missing stripes if activating an incomplete volume.
    # Using "error" will make inaccessible parts of the device return
    # I/O errors on access. You can instead use a device path, in which
    # case, that device will be used to in place of missing stripes.
    # But note that using anything other than "error" with mirrored
    # or snapshotted volumes is likely to result in data corruption.
    missing_stripe_filler = "error"

    # How much stack (in KB) to reserve for use while devices suspended
    reserved_stack = 256

    # How much memory (in KB) to reserve for use while devices suspended
    reserved_memory = 8192

    # Nice value used while devices suspended
    process_priority = -18

    # If volume_list is defined, each LV is only activated if there is a
    # match against the list.
    # "vgname" and "vgname/lvname" are matched exactly.
    # "@tag" matches any tag set in the LV or VG.
    # "@*" matches if any tag defined on the host is also set in the LV
    or VG
    #
    # volume_list = [ "vg1", "vg2/lvol1", "@tag1", "@*" ]

    # Size (in KB) of each copy operation when mirroring
    mirror_region_size = 512

```

```

# Setting to use when there is no readahead value stored in the
metadata.
#
# "none" - Disable readahead.
# "auto" - Use default value chosen by kernel.
readahead = "auto"

# 'mirror_image_fault_policy' and 'mirror_log_fault_policy' define
# how a device failure affecting a mirror is handled.
# A mirror is composed of mirror images (copies) and a log.
# A disk log ensures that a mirror does not need to be re-synced
# (all copies made the same) every time a machine reboots or crashes.
#
# In the event of a failure, the specified policy will be used to
determine
# what happens. This applies to automatic repairs (when the mirror is
being
# monitored by dmeventd) and to manual lvconvert --repair when
# --use-policies is given.
#
# "remove" - Simply remove the faulty device and run without it. If
#             the log device fails, the mirror would convert to using
#             an in-memory log. This means the mirror will not
#             remember its sync status across crashes/reboots and
#             the entire mirror will be re-synced. If a
#             mirror image fails, the mirror will convert to a
#             non-mirrored device if there is only one remaining good
#             copy.
#
# "allocate" - Remove the faulty device and try to allocate space on
#               a new device to be a replacement for the failed device.
#               Using this policy for the log is fast and maintains the
#               ability to remember sync state through crashes/reboots.
#               Using this policy for a mirror device is slow, as it
#               requires the mirror to resynchronize the devices, but it
#               will preserve the mirror characteristic of the device.
#               This policy acts like "remove" if no suitable device and
#               space can be allocated for the replacement.
#
# "allocate_anywhere" - Not yet implemented. Useful to place the log
device
#                       temporarily on same physical volume as one of the mirror
#                       images. This policy is not recommended for mirror devices
#                       since it would break the redundant nature of the mirror.
This
#                       policy acts like "remove" if no suitable device and space
can
#                       be allocated for the replacement.

mirror_log_fault_policy = "allocate"
mirror_image_fault_policy = "remove"

# 'snapshot_autoextend_threshold' and 'snapshot_autoextend_percent'
define
# how to handle automatic snapshot extension. The former defines when

```

```

the
    # snapshot should be extended: when its space usage exceeds this many
    # percent. The latter defines how much extra space should be allocated
for
    # the snapshot, in percent of its current size.
    #
    # For example, if you set snapshot_autoextend_threshold to 70 and
    # snapshot_autoextend_percent to 20, whenever a snapshot exceeds 70%
usage,
    # it will be extended by another 20%. For a 1G snapshot, using up 700M
will
    # trigger a resize to 1.2G. When the usage exceeds 840M, the snapshot
will
    # be extended to 1.44G, and so on.
    #
    # Setting snapshot_autoextend_threshold to 100 disables automatic
    # extensions. The minimum value is 50 (A setting below 50 will be
treated
    # as 50).

    snapshot_autoextend_threshold = 100
    snapshot_autoextend_percent = 20

    # While activating devices, I/O to devices being (re)configured is
    # suspended, and as a precaution against deadlocks, LVM2 needs to pin
    # any memory it is using so it is not paged out. Groups of pages that
    # are known not to be accessed during activation need not be pinned
    # into memory. Each string listed in this setting is compared against
    # each line in /proc/self/maps, and the pages corresponding to any
    # lines that match are not pinned. On some systems locale-archive was
    # found to make up over 80% of the memory used by the process.
    # mlock_filter = [ "locale/locale-archive", "gconv/gconv-
modules.cache" ]

    # Set to 1 to revert to the default behaviour prior to version 2.02.62
    # which used mlockall() to pin the whole process's memory while
activating
    # devices.
    use_mlockall = 0

    # Monitoring is enabled by default when activating logical volumes.
    # Set to 0 to disable monitoring or use the --ignoremonitoring option.
    monitoring = 1

    # When pvmove or lvconvert must wait for the kernel to finish
    # synchronising or merging data, they check and report progress
    # at intervals of this number of seconds. The default is 15 seconds.
    # If this is set to 0 and there is only one thing to wait for, there
    # are no progress reports, but the process is awoken immediately the
    # operation is complete.
    polling_interval = 15
}

#####
# Advanced section #

```

```
#####
```

```
# Metadata settings
```

```
#
```

```
# metadata {
```

```
    # Default number of copies of metadata to hold on each PV.  0, 1 or 2.
    # You might want to override it from the command line with 0
    # when running pvcreate on new PVs which are to be added to large VGs.
```

```
    # pvmetadatasize = 1
```

```
    # Default number of copies of metadata to maintain for each VG.
    # If set to a non-zero value, LVM automatically chooses which of
    # the available metadata areas to use to achieve the requested
    # number of copies of the VG metadata.  If you set a value larger
    # than the the total number of metadata areas available then
    # metadata is stored in them all.
    # The default value of 0 ("unmanaged") disables this automatic
    # management and allows you to control which metadata areas
    # are used at the individual PV level using 'pvchange
    # --metadataignore y/n'.
```

```
    # vgmetadatasize = 0
```

```
    # Approximate default size of on-disk metadata areas in sectors.
    # You should increase this if you have large volume groups or
    # you want to retain a large on-disk history of your metadata changes.
```

```
    # pvmetadatasize = 255
```

```
    # List of directories holding live copies of text format metadata.
    # These directories must not be on logical volumes!
    # It's possible to use LVM2 with a couple of directories here,
    # preferably on different (non-LV) filesystems, and with no other
    # on-disk metadata (pvmetadatasize = 0). Or this can be in
    # addition to on-disk metadata areas.
    # The feature was originally added to simplify testing and is not
    # supported under low memory situations - the machine could lock up.
    #
    # Never edit any files in these directories by hand unless you
    # you are absolutely sure you know what you are doing! Use
    # the supplied toolset to make changes (e.g. vgcfgrestore).
```

```
    # dirs = [ "/etc/lvm/metadata", "/mnt/disk2/lvm/metadata2" ]
```

```
#}
```

```
# Event daemon
```

```
#
```

```
dmeventd {
```

```
    # mirror_library is the library used when monitoring a mirror device.
    #
    # "libdevmapper-event-lvm2mirror.so" attempts to recover from
    # failures.  It removes failed devices from a volume group and
    # reconfigures a mirror as necessary.  If no mirror library is
    # provided, mirrors are not monitored through dmeventd.
```

```
mirror_library = "libdevmapper-event-lvm2mirror.so"

# snapshot_library is the library used when monitoring a snapshot
device.
#
# "libdevmapper-event-lvm2snapshot.so" monitors the filling of
# snapshots and emits a warning through syslog when the use of
# the snapshot exceeds 80%. The warning is repeated when 85%, 90% and
# 95% of the snapshot is filled.

snapshot_library = "libdevmapper-event-lvm2snapshot.so"

# Full path of the dmeventd binary.
#
# executable = "/sbin/dmeventd"
}
```

APPENDICE C. TAG OGGETTO LVM

Un tag LVM può essere usato per raggruppare gli oggetti LVM2 dello stesso tipo. I tag possono essere collegati ad oggetti come ad esempio i volumi fisici, gruppi di volumi, volumi logici e segmenti. Essi possono essere collegati agli host in una configurazione cluster. Non è possibile creare un tag per le snapshot.

È possibile usare i tag sulla linea di comando al posto di PV, VG o argomenti di LV. I suddetti tag devono avere come prefisso @ per evitare qualsiasi incomprensione. Ogni tag viene esteso tramite la sostituzione con oggetti che possiedono lo stesso tag e di tipo previsto dalla posizione sulla linea di comando.

Con la release Red Hat Enterprise Linux 6.1 i tag di LVM sono stringhe composte da un massimo di 1024 caratteri (per release precedenti il limite massimo era di 128 caratteri). I suddetti tag non possono iniziare con un trattino.

Un tag valido può consistere solo di una gamma di caratteri limitata. Per la release di Red Hat Enterprise Linux 6.0 i caratteri permessi sono [A-Za-z0-9_+.-]. Con Red Hat Enterprise Linux 6.1 l'elenco di caratteri permessi è stato esteso ed ora comprende "/", "=", "!", ":", "#", and "&".

È possibile etichettare solo gli oggetti presenti in un gruppo di volumi. I volumi fisici perdono i propri tag se rimossi da un gruppo di volumi; ciò si verifica poichè i tag vengono archiviati come parte dei metadati del gruppo di volumi, e cancellati quando un volume fisico viene rimosso. Le snapshot non possono essere etichettate.

Il seguente comando elenca tutti i volumi logici con il tag **database**.

```
lvs @database
```

C.1. COME AGGIUNGERE E RIMUOVERE I TAG DAGLI OGGETTI

Per aggiungere o rimuovere tag dai volumi fisici utilizzate l'opzione **--addtag** o **--deltag** del comando **pvchange**.

Per aggiungere o cancellare i tag dai gruppi di volumi utilizzate l'opzione **--addtag** o **--deltag** dei comandi **vgchange** o **vgcreate**.

Per aggiungere o cancellare i tag dai volumi logici, utilizzate l'opzione **--addtag** o **--deltag** dei comandi **lvchange** o **lvcreate**.

Con la release Red Hat Enterprise Linux 6.1 è possibile specificare argomenti **--addtag** e **--deltag** multipli all'interno di un comando **pvchange**, **vgchange**, o **lvchange**. Per esempio il seguente comando cancella i tag **T9** e **T10** aggiungendo **T13** e **T14** al gruppo di volumi **grant**.

```
vgchange --deltag T9 --deltag T10 --addtag T13 --addtag T14 grant
```

C.2. TAG DELL'HOST

In una configurazione del cluster è possibile definire i tag dell'host nei file di configurazione. Se impostate **hosttags = 1** nella sezione **tags**, verrà definito automaticamente un tag dell'host utilizzando l'hostname della macchina. Tale processo vi permetterà di usare un file di configurazione il quale può essere replicato su tutte le vostre macchine in modo da mantenere copie identiche del file, pur avendo un comportamento diverso in base all'hostname.

Per maggiori informazioni sui file di configurazione consultate [Appendice B, File di configurazione LVM](#).

Per ogni tag verrà letto un file di configurazione aggiuntivo se esistente: `lvm_hosttag.conf`. Se il file in questione definisce nuovi tag, allora verranno aggiunti all'elenco nuovi file di configurazione da leggere.

Per esempio, la seguente voce presente nel file di configurazione definisce sempre **tag1**, e definisce **tag2** se l'hostname è **host1**.

```
tags { tag1 { } tag2 { host_list = ["host1"] } }
```

C.3. CONTROLLO ATTIVAZIONE CON I TAG

È possibile specificare all'interno del file di configurazione che solo determinati volumi logici possono essere attivati sull'host desiderato. Per esempio, la seguente voce si comporta come un filtro per le richieste di attivazione (come ad esempio **vgchange -ay**), attivando solo **vg1/lvol0** e qualsiasi altro volume logico o gruppo di volumi con tag **database** nei metadati sull'host interessato.

```
activation { volume_list = ["vg1/lvol0", "@database" ] }
```

È presente una corrispondenza speciale "@*", che determina una corrispondenza solo se qualsiasi tag dei metadati corrisponde al tag di un host su quella macchina.

In un altro esempio considerate una situazione dove ogni macchina nel cluster possiede le seguenti voci nel file di configurazione:

```
tags { hosttags = 1 }
```

Se desiderate attivare **vg1/lvol2** solo sull'host **db2**, fate quanto segue:

1. Eseguite **lvchange --addtag @db2 vg1/lvol2** da qualsiasi host nel cluster.
2. Eseguite **lvchange -ay vg1/lvol2**.

Questa soluzione comporta la conservazione degli hostname all'interno dei metadati del gruppo di volumi.

APPENDICE D. METADATI DEL GRUPPO DI VOLUMI LVM

Le informazioni relative alla configurazione di un gruppo di volumi vengono chiamate metadati. Per default, una copia identica di metadati viene mantenuta nelle aree riservate ai metadati in ogni volume fisico all'interno del gruppo di volumi. I metadati del gruppo di volumi LVM sono di piccole dimensioni e archiviati come ASCII.

Se un gruppo di volumi contiene un certo numero di volumi fisici, avere un numero elevato di copie ridondanti di metadati non è conveniente. È possibile creare un volume fisico senza avere alcuna copia di metadati tramite l'opzione `--metadatacopies 0` del comando `pvcreate`. Una volta selezionato il numero di copie dei metadati che il volume fisico deve contenere, tale valore non potrà essere più modificato in futuro. Selezionando 0 si avranno aggiornamenti più veloci delle modifiche relative alla configurazione. Da notare tuttavia che in ogni momento il gruppo di volumi deve contenere almeno un volume fisico con un'area di metadati (se non state utilizzando impostazioni avanzate di configurazione che vi permetteranno di conservare i metadati del gruppo di volumi in un file system). Se in futuro desiderate dividere il gruppo di volumi ogni gruppo avrà bisogno di almeno una copia di metadati.

I metadati principali vengono archiviati in ASCII. L'area per i metadati è un buffer circolare. I nuovi metadati saranno aggiunti a quelli più vecchi e successivamente verrà aggiornato il puntatore.

È possibile specificare la dimensione dell'area dei metadati con l'opzione `--metadatasize` del comando `pvcreate`. La dimensione predefinita è troppo piccola per i gruppi di volumi con un numero elevato di volumi logici o volumi fisici.

D.1. ETICHETTA DEL VOLUME FISICO

Per default il comando `pvcreate` posiziona l'etichetta del volume fisico nel secondo settore di 512-byte. La suddetta etichetta può essere posizionata in qualsiasi dei primi quattro settori, poichè i tool di LVM alla ricerca di una etichetta del volume fisico controllano i primi quattro settori. L'etichetta del volume fisico inizia con la stringa **LABELONE**.

L'etichetta del volume fisico contiene:

- UUID del volume fisico
- La dimensione del dispositivo a blocchi in byte
- Un elenco terminato da NULL delle posizioni dell'area dei dati
- Elenchi terminati da NULL di posizioni dell'area dei metadati

Le posizioni dei metadati sono conservate come offset e dimensioni (in byte). All'interno dell'etichetta vi è spazio disponibile per 15 posizioni, ma i tool di LVM ne utilizzano solo 3: un'area di dati singola più un massimo di due aree di metadati.

D.2. CONTENUTI DEI METADATI

I metadati del gruppo di volumi contengono:

- Informazioni su quando e come sono stati creati
- Informazioni sul gruppo di volumi:

Le informazioni del gruppo di volumi contengono:

- Nome ed id unico
- Un numero della versione il quale viene incrementato ogni qualvolta vengono aggiornati i metadati
- Proprietà: Lettura/Scrittura? Ridimensionabile?
- Qualsiasi limite amministrativo sul numero di volume logico/fisico che si possono contenere
- L'entità della dimensione (in unità di secondi definiti come 512 byte)
- Un elenco non ordinato di volumi fisici che costituiscono il gruppo di volumi, ognuno con:
 - Il proprio UUID, viene usato per determinare il dispositivo a blocchi che lo contiene
 - Qualsiasi proprietà, come ad esempio se il volume fisico è assegnabile
 - L'offset per l'inizio della prima estensione all'interno del volume fisico (in settori)
 - Il numero delle estensioni
- Un elenco non ordinato di volumi logici. Ognuno costituito da
 - Un elenco ordinato di segmenti di volume logico. Per ogni segmento i metadati includono una mappatura applicata ad un elenco ordinato di segmenti del volume fisico o segmenti di volume logico

D.3. ESEMPIO DI METADATI

Quanto segue mostra un esempio di metadati del gruppo di volumi LVM per un gruppo di volumi chiamato **myvg**.

```
# Generated by LVM2: Tue Jan 30 16:28:15 2007

contents = "Text Format Volume Group"
version = 1

description = "Created *before* executing 'lvextend -L+5G /dev/myvg/mylv
/dev/sdc'"

creation_host = "tng3-1"           # Linux tng3-1 2.6.18-8.el5 #1 SMP Fri Jan
26 14:15:21 EST 2007 i686
creation_time = 1170196095         # Tue Jan 30 16:28:15 2007

myvg {
    id = "0zd3UT-wbYT-lDHq-lMPs-EjoE-0o18-wL28X4"
    seqno = 3
    status = ["RESIZEABLE", "READ", "WRITE"]
    extent_size = 8192              # 4 Megabytes
    max_lv = 0
    max_pv = 0

    physical_volumes {
        pv0 {
            id = "ZBW5qW-dXF2-0bGw-ZCad-2RlV-phwu-1c1RFt"
```

```

        device = "/dev/sda"      # Hint only

        status = ["ALLOCATABLE"]
        dev_size = 35964301      # 17.1491 Gigabytes
        pe_start = 384
        pe_count = 4390 # 17.1484 Gigabytes
    }

    pv1 {
        id = "ZHEZJW-MR64-D3QM-Rv7V-Hxsa-zU24-wztY19"
        device = "/dev/sdb"      # Hint only

        status = ["ALLOCATABLE"]
        dev_size = 35964301      # 17.1491 Gigabytes
        pe_start = 384
        pe_count = 4390 # 17.1484 Gigabytes
    }

    pv2 {
        id = "wCoG4p-55Ui-9tbp-VTEA-j06s-RAVx-UREW0G"
        device = "/dev/sdc"      # Hint only

        status = ["ALLOCATABLE"]
        dev_size = 35964301      # 17.1491 Gigabytes
        pe_start = 384
        pe_count = 4390 # 17.1484 Gigabytes
    }

    pv3 {
        id = "hGlUwi-zsBg-39FF-do88-pHxY-8XA2-9WKIiA"
        device = "/dev/sdd"      # Hint only

        status = ["ALLOCATABLE"]
        dev_size = 35964301      # 17.1491 Gigabytes
        pe_start = 384
        pe_count = 4390 # 17.1484 Gigabytes
    }
}
logical_volumes {

    mylv {
        id = "GhUYSF-qVM3-rzQo-a6D2-o0aV-LQet-Ur90F9"
        status = ["READ", "WRITE", "VISIBLE"]
        segment_count = 2

        segment1 {
            start_extent = 0
            extent_count = 1280      # 5 Gigabytes

            type = "striped"
            stripe_count = 1          # linear

            stripes = [
                "pv0", 0
            ]
        }
    }
}

```

```

    }
    }
    }
    segment2 {
        start_extent = 1280
        extent_count = 1280      # 5 Gigabytes

        type = "striped"
        stripe_count = 1        # linear

        stripes = [
            "pv1", 0
        ]
    }
}
}
```

APPENDICE E. CRONOLOGIA DI REVISIONE

Revisione 1-5.400 Rebuild with publican 4.0.0	2013-10-31	Rüdiger Landmann
---	-------------------	-------------------------

Revisione 1-5 Rebuild for Publican 3.0	2012-07-18	Anthony Towns
--	-------------------	----------------------

Revisione 2.0-1 Release iniziale per Red Hat Enterprise Linux 6.1	Thu May 19 2011	Steven Levine
---	------------------------	----------------------

Risolve: #694619

Documenta una nuova politica di assegnazione **cling** quando si estende un volume logico.

Risolve: #682649

Aggiunge un avvertimento sull'esecuzione di comandi multipli per la creazione del mirror in successione su volumi clusterizzati.

Risolve: #674100

Aggiunge un esempio di output per il comando **dmsetup ls --tree**.

Risolve: #694607

Documenta il supporto per l'inclusione di argomenti multipli **--addtag** e **--deltag** su di una linea di comando.

Risolve: #694604

Documenta il supporto per l'elenco di caratteri estesi con i tag.

Risolve: #694611

Documenta il supporto dei mirror segmentati.

Risolve: #694616

Documenta il supporto per le snapshot dei volumi speculari.

Risolve: #694618

Documenta il supporto per le snapshot di volumi cluster esclusivamente-attivati.

Risolve: #682648

Documenta che quando un segmento del mirror viene riassegnato anche il log del mirror potrà essere spostato.

Risolve: #661530

Aggiorna l'esempio **cluster.conf** per documentare le funzioni correnti.

Risolve: #642400

Aggiunge una nota relativa alla gestione del log del cluster da parte del nodo con l'ID più basso.

Risolve: #663462

Rimuove i riferimenti non aggiornati per il monitor della macchina virtuale Xen.

Revisione 1.0-1 Release iniziale per Red Hat Enterprise Linux 6	Wed Nov 10 2010	Steven Levine
---	------------------------	----------------------

INDICE ANALITICO

Simboli

comando lvchange , [Modifica dei parametri di un gruppo di volumi logici](#)

comando lvdisplay , [Visualizzazione dei volumi logici](#)

comando vgscan , [Scansione dischi per i gruppi di volumi per la creazione del file di cache](#)

A

ambiente cluster, [LVM Logical Volume Manager \(CLVM\)](#) , [Creazione dei volumi LVM in un cluster](#)
assegnazione

politica, [Creazione dei gruppi di volumi](#)

prevenzione, [Come impedire l'assegnazione su di un volume fisico](#)

attivazione gruppi di volumi, [Attivazione e disattivazione dei gruppi di volumi](#)

nodi individuali, [Attivazione e disattivazione dei gruppi di volumi](#)

solo nodo locale, [Attivazione e disattivazione dei gruppi di volumi](#)

attivazione volumi logici

nodi individuali, [Attivazione dei volumi logici su nodi individuali in un cluster](#)

B

backup

file, [Backup del volume logico](#)

metadati, [Backup del volume logico](#) , [Esecuzione del back up dei metadati del gruppo di volumi](#)

C

CLVM

definizione, [LVM Logical Volume Manager \(CLVM\)](#)

comando lvconvert , [Come modificare la configurazione del volume speculare](#)

comando lvcreate , [Creazione di volumi logici lineari](#)

comando lvextend , [Come aumentare la dimensione dei volumi logici](#)

comando lvmdiskscan , [Scansione per dispositivi a blocchi](#)

comando lvreduce , [Modifica della dimensione dei volumi logici](#) , [Come ridurre la dimensione dei volumi logici](#)

comando lvremove , [Rimozione dei volumi logici](#)

comando lvrename , [Modifica del nome dei volumi logici](#)

comando lvs , [Personalizzazione dei riporti per LVM](#) , [Il comando lvs](#)

opzioni di visualizzazione, [Il comando lvs](#)

comando lvscan , [Visualizzazione dei volumi logici](#)

comando pvdisplay , [Visualizzazione dei volumi fisici](#)

comando pvmove , [Riposizionamento dati online](#)

comando pvremove , [Rimozione dei volumi fisici](#)

comando pvresize , [Come variare la dimensione di un volume fisico](#)

comando pvs , [Personalizzazione dei riporti per LVM](#)

opzioni di visualizzazione, [Il comando pvs](#)

comando pvscan , [Visualizzazione dei volumi fisici](#)

comando vgcfbackup , [Esecuzione del back up dei metadati del gruppo di volumi](#)

comando vgcfrestore , [Esecuzione del back up dei metadati del gruppo di volumi](#)

comando vgchange , [Modifica dei parametri di un gruppo di volumi](#)

comando vgcreate , [Creazione dei gruppi di volumi](#) , [Creazione dei gruppi di volumi in un cluster](#)

comando vgdisplay , [Come visualizzare i gruppi di volumi](#)

comando vgexport , [Come spostare un gruppo di volumi su di un altro sistema](#)

comando vgextend , [Aggiunta di un volume fisico ad un gruppo di volumi](#)

comando vgimport , [Come spostare un gruppo di volumi su di un altro sistema](#)

comando vgmerge , [Come unire i gruppi di volumi](#)

comando vgmknodes , [Come ricreare una directory del gruppo di volumi](#)

comando vgreduce , [Rimozione dei volumi fisici da un gruppo di volumi](#)

comando vgrename , [Come rinominare un gruppo di volumi](#)

comando vgs , [Personalizzazione dei riporti per LVM](#)

opzioni di visualizzazione, [Il comando vgs](#)

comando vgsplit , [Separazione di un gruppo di volumi](#)

commento

informazioni di contatto per questo manuale, [Abbiamo bisogno di commenti!](#)

creazione

gruppi di volumi, [Creazione dei gruppi di volumi](#)

gruppo di volumi, clusterizzato, [Creazione dei gruppi di volumi in un cluster](#)

volume logico, [Creazione di volumi logici lineari](#)

volume logico segmentato, esempio, [Creazione di un volume logico segmentato](#)

volume logico, esempio, [Creazione di un volume logico LVM su tre dischi](#)

volumi fisici, [Creazione dei volumi fisici](#)

volumi LVM in un cluster, [Creazione dei volumi LVM in un cluster](#)

creazione volumi LVM

panoramica, [Panoramica sulla creazione del volume logico](#)

D

demone clvmd , [LVM Logical Volume Manager \(CLVM\)](#)

device manager udev, [Supporto Device Mapper per il Device Manager udev](#)

dimensione dispositivo, massima, [Creazione dei gruppi di volumi](#)

directory /lib/udev/rules.d, [Integrazione udev con il Device Mapper](#)

directory del file speciale del dispositivo, [Creazione dei gruppi di volumi](#)

directory rules.d, [Integrazione udev con il Device Mapper](#)

disattivazione gruppi di volumi, [Attivazione e disattivazione dei gruppi di volumi](#)

escluso su di un nodo, [Attivazione e disattivazione dei gruppi di volumi](#)

solo nodo locale, [Attivazione e disattivazione dei gruppi di volumi](#)

display d'aiuto, [Come utilizzare i comandi CLI](#)

display pagina man, [Come utilizzare i comandi CLI](#)

dispositivi falliti

visualizzazione, [Come visualizzare le informazioni su dispositivi falliti](#)

dispositivo a blocchi

scansione, [Scansione per dispositivi a blocchi](#)

E

esempi di configurazione, [Esempi di configurazione LVM](#)

estensione

assegnazione, [Creazione dei gruppi di volumi](#)

definizione, [Gruppi di volumi](#), [Creazione dei gruppi di volumi](#)

estensione fisica

prevenzione dell'assegnazione, [Come impedire l'assegnazione su di un volume fisico](#)

F

file cache

compilazione, [Scansione dischi per i gruppi di volumi per la creazione del file di cache](#)

file d'archivio, [Backup del volume logico](#), [Esecuzione del back up dei metadati del gruppo di volumi](#)

file di backup, [Esecuzione del back up dei metadati del gruppo di volumi](#)

file system

sviluppo su di un volume logico, [Sviluppo di un file system su di un volume logico](#)

filtri, [Controllo delle scansioni del dispositivo LVM con i filtri](#)

filtri scansione del dispositivo, [Controllo delle scansioni del dispositivo LVM con i filtri](#)

formato riporto, dispositivi LVM, [Personalizzazione dei riporti per LVM](#)

funzioni, nuove e modificate, [Funzioni nuove e modificate](#)

G

gruppo di volumi

amministrare, generale, [Amministrazione del gruppo di volumi](#)

attivazione, [Attivazione e disattivazione dei gruppi di volumi](#)

aumento, [Aggiunta di un volume fisico ad un gruppo di volumi](#)

come spostarsi tra sistemi, [Come spostare un gruppo di volumi su di un altro sistema](#)

creazione, [Creazione dei gruppi di volumi](#)

creazione in un cluster, [Creazione dei gruppi di volumi in un cluster](#)

definizione, [Gruppi di volumi](#)

disattivazione, [Attivazione e disattivazione dei gruppi di volumi](#)

divisione, [Separazione di un gruppo di volumi](#)

estensione, [Aggiunta di un volume fisico ad un gruppo di volumi](#)

modifica dei parametri, [Modifica dei parametri di un gruppo di volumi](#)

opzioni di visualizzazione vgs , [Il comando vgs](#)

riduzione, [Rimozione dei volumi fisici da un gruppo di volumi](#)

rimozione, [Rimozione dei gruppi di volumi](#)

rinominare, [Come rinominare un gruppo di volumi](#)

separazione

procedura di esempio, [Separazione di un gruppo di volumi](#)

unione, [Come unire i gruppi di volumi](#)

visualizzazione, [Come visualizzare i gruppi di volumi](#) , [Personalizzazione dei riporti per LVM](#) , [Il comando vgs](#)

I

inizializzazione

partizioni, [Inizializzazione dei volumi fisici](#)

volumi fisici, [Inizializzazione dei volumi fisici](#)

L

LVM

aiuto, [Come utilizzare i comandi CLI](#)

amministrazione volume fisico, [Amministrazione del volume fisico](#)

amministrazione volume logico, [Amministrazione del volume logico](#)

clusterizzato, [LVM Logical Volume Manager \(CLVM\)](#)

componenti, [Panoramica sull'architettura LVM](#) , [Componenti di LVM](#)

cronologia degli eventi, [Panoramica sull'architettura LVM](#)

etichetta, [Volumi fisici](#)

formato riporto personalizzato, [Personalizzazione dei riporti per LVM](#)

gruppo di volumi, definizione, [Gruppi di volumi](#)

panoramica sull'architettura, [Panoramica sull'architettura LVM](#)

registrazione, [Registrazione](#)

struttura della directory, [Creazione dei gruppi di volumi](#)

volume fisico, definizione, [Volumi fisici](#)

LVM1, [Panoramica sull'architettura LVM](#)

LVM2, [Panoramica sull'architettura LVM](#)

M

Messaggio di estensioni libere insufficienti, [Estensioni disponibili insufficienti per un volume logico](#)
metadati

backup, [Backup del volume logico](#), [Esecuzione del back up dei metadati del gruppo di volumi](#)
ripristino, [Recupero dei metadati del volume fisico](#)

modifica nome

volume logico, [Modifica del nome dei volumi logici](#)

N

nomi percorso, [Come utilizzare i comandi CLI](#)

nomi percorso del dispositivo, [Come utilizzare i comandi CLI](#)

numeri del dispositivo

maggiore, [Numeri del dispositivo persistenti](#)

minore, [Numeri del dispositivo persistenti](#)

persistenti, [Numeri del dispositivo persistenti](#)

numeri del dispositivo persistenti, [Numeri del dispositivo persistenti](#)

O

output verboso, [Come utilizzare i comandi CLI](#)

P

panoramica

funzioni, nuove e modificate, [Funzioni nuove e modificate](#)

parametro di configurazione mirror_image_fault_policy, [Politica sugli errori del volume logico](#)
[speculare](#)

parametro di configurazione mirror_log_fault_policy, [Politica sugli errori del volume logico](#)
[speculare](#)

partizioni

multiple, [Partizioni multiple su di un disco](#)

procedure amministrative, [Panoramica sull'amministrazione di LVM](#)

R

registrazione, [Registrazione](#)

regole udev, [Integrazione udev con il Device Mapper](#)

ridimensionamento

volume fisico, [Come variare la dimensione di un volume fisico](#)

volume logico, [Modifica della dimensione dei volumi logici](#)

rimozione

disco da un volume logico, [Rimozione di un disco da un volume logico](#)

volume logico, [Rimozione dei volumi logici](#)

volumi fisici, [Rimozione dei volumi fisici](#)

rinominare

gruppo di volumi, [Come rinominare un gruppo di volumi](#)

riposizionamento dati online, [Riposizionamento dati online](#)

riposizionamento dati, online, [Riposizionamento dati online](#)

S

scansione

dispositivi a blocchi, [Scansione per dispositivi a blocchi](#)

scansione dispositivi, filtri, [Controllo delle scansioni del dispositivo LVM con i filtri](#)

snapshot del volume

definizione, [Volumi delle snapshot](#)

sviluppo di un file system

volume logico, [Sviluppo di un file system su di un volume logico](#)

T

tipo di partizione, impostazione, [Impostazione del tipo di partizione](#)

troubleshooting, [Troubleshooting di LVM](#)

U

unità linea di comando, [Come utilizzare i comandi CLI](#)

unità, linea di comando, [Come utilizzare i comandi CLI](#)

V

visualizza

come ordinare l'output, [Come ordinare i riporti di LVM](#)

visualizzazione

gruppi di volumi, [Come visualizzare i gruppi di volumi](#), [Il comando vgs](#)

volumi fisici, [Visualizzazione dei volumi fisici](#), [Il comando pvs](#)

volumi logici, [Visualizzazione dei volumi logici](#), [Il comando lvs](#)

volume fisico

aggiunta ad un gruppo di volumi, [Aggiunta di un volume fisico ad un gruppo di volumi](#)

amministrazione, generale, [Amministrazione del volume fisico](#)

creazione, [Creazione dei volumi fisici](#)

definizione, [Volumi fisici](#)

display, [Il comando pvs](#)

disposizione, [Disposizione del volume fisico LVM](#)

illustrazione, [Disposizione del volume fisico LVM](#)

inizializzazione, [Inizializzazione dei volumi fisici](#)
opzioni di visualizzazione pvs , [Il comando pvs](#)
ridimensionamento, [Come variare la dimensione di un volume fisico](#)
rimozione, [Rimozione dei volumi fisici](#)
rimozione da un gruppo di volumi, [Rimozione dei volumi fisici da un gruppo di volumi](#)
rimozione volume perso, [Rimozione dei volumi fisici persi da un gruppo di volumi](#)
ripristino, [Sostituzione di un volume fisico mancante](#)
visualizzazione, [Visualizzazione dei volumi fisici](#) , [Personalizzazione dei riporti per LVM](#)

volume logico

accesso esclusivo, [Attivazione dei volumi logici su nodi individuali in un cluster](#)
accesso locale, [Attivazione dei volumi logici su nodi individuali in un cluster](#)
amministrare, generale, [Amministrare del volume logico](#)
aumento, [Come aumentare la dimensione dei volumi logici](#)
creazione, [Creazione di volumi logici lineari](#) , [Creazione dei volumi segmentati](#) , [Creazione volumi speculari](#)
definizione, [Volumi logici](#), [Volumi logici LVM](#)
esempio creazione, [Creazione di un volume logico LVM su tre dischi](#)
estensione, [Come aumentare la dimensione dei volumi logici](#)
lineare, [Creazione di volumi logici lineari](#)
modifica dei parametri, [Modifica dei parametri di un gruppo di volumi logici](#)
modifica nome, [Modifica del nome dei volumi logici](#)
opzioni di visualizzazione lvs , [Il comando lvs](#)
ridimensionamento, [Modifica della dimensione dei volumi logici](#)
riduzione, [Come ridurre la dimensione dei volumi logici](#)
rimozione, [Rimozione dei volumi logici](#)
segmentato, [Creazione dei volumi segmentati](#)
snapshot, [Creazione dei volumi della snapshot](#)
speculare, [Creazione volumi speculari](#)
visualizzazione, [Visualizzazione dei volumi logici](#) , [Personalizzazione dei riporti per LVM](#) , [Il comando lvs](#)

volume logico della snapshot

creazione, [Creazione dei volumi della snapshot](#)

volume logico lineare

conversione a speculare, [Come modificare la configurazione del volume speculare](#)
creazione, [Creazione di volumi logici lineari](#)
definizione, [Volumi lineari](#)

volume logico segmentato

aumento, [Come estendere un volume segmentato](#)
definizione, [Volumi logici segmentati](#)
esempio di creazione, [Creazione di un volume logico segmentato](#)

estensione, [Come estendere un volume segmentato](#)

volume logico speculare

clusterizzato, [Creazione di un volume logico LVM speculare in un cluster](#)

conversione a lineare, [Come modificare la configurazione del volume speculare](#)

definizione, [Volumi logici speculari](#)

politica sugli errori, [Politica sugli errori del volume logico speculare](#)

riconfigurazione, [Come modificare la configurazione del volume speculare](#)

ripristino da un errore, [Processo di recupero da un LVM Mirror Failure](#)