Guide pratique de LVM

Version française du *LVM HOWTO*

A. J. Lewis

Guide pratique de LVM : Version française du *LVM HOW-TO*

A. J. Lewis
Adaptation française: Emmanuel Araman
Relecture de la version française: Camille Huot, Alain Portal
Préparation de la publication de la v.f.: Jean-Philippe Guérard

Publié 11 octobre 2004

Résumé

Ce document explique comment compiler, installer et configurer le gestionnaire de volumes logiques (LVM) de Linux. Il inclut aussi une description synthétique de LVM. Cette version de guide pratique couvre LVM 2 avec le mappeur de périphériques et LVM 1.0.8.

Table des matières

Introduction

Ce document essaie de rassembler tous les éléments nécessaires pour obtenir LVM et le faire fonctionner. Il traite l'ensemble du processus pour l'obtenir, le compiler, l'installer et le mettre en œuvre. Il fournit aussi des liens vers des configurations testées de LVM. Cette version de guide pratique couvre LVM 2 et LVM 1.0.8.

Toutes les versions précédentes de LVM sont considérées comme obsolètes et sont gardées uniquement pour des raisons historiques. Ce document n'a pas pour objet d'essayer d'expliquer et décrire le fonctionnement de ces versions.

1. Dernière version de ce document

Vous trouverez la plus récente version française de ce document à l'adresse : http://www.traduc.org/docs/howto/lecture/LVM-HOWTO.html.

La dernière version originale de ce guide pratique est disponible sur le serveur CVS du Projet de documentation Linux avec les autres guides pratiques (*howto*). Vous pouvez l'obtenir en faisant un *checkout* de « LDP/howto/docbook/LVM-HOWTO.xml » sur le serveur CVS du LDP. Vous devriez toujours pouvoir obtenir une version lisible de ce guide pratique à l'adresse : http://www.tldp.org/HOWTO/LVM-HOWTO.html.

2. Limitations de responsabilité

3. Auteurs

Liste de tous ceux qui ont écrit dans ce document.

- AJ Lewis [mailto:alewis CHEZ redhat POINT com_NOSPAM]
- Joe Thornber [mailto:thornber CHEZ redhat POINT com_NOSPAM]
- Patrick Caulfield [mailto:pcaulfie CHEZ redhat POINT com_NOSPAM]
- Alasdair Kergon [mailto:agk CHEZ redhat POINT com_NOSPAM]
- Jochen Radmacher [mailto:jradmacher CHEZ gmx POINT de_NOSPAM] - Informations sur l'extension de JFS

S'il vous plaît, veuillez prévenir le mainteneur du guide pratique si vous pensez devoir être dans la liste ci-dessus.

Chapitre 1. Qu'est-ce que LVM ?

LVM est un gestionnaire de volumes logiques (Logical Volume Manager) pour le système d'exploitation Linux. Il existe désormais deux versions de LVM pour Linux :

- LVM 2 - La dernière et meilleure version de LVM pour Linux.

  LVM 2 est presque entièrement compatible avec les volumes créés avec LVM 1, à l'exception des instantanés (il faut supprimer les volumes d'instantanés avant de passer à LVM 2).

  LVM 2 utilise le mappeur de périphériques du noyau Linux. Le mappeur de périphériques est inclus dans les noyaux 2.6. Des mises à jour sont disponibles pour les noyaux 2.4 récents.

- LVM 1 - La version présente dans les noyaux 2.4.

  LVM 1 est un produit mature et considéré comme stable depuis plusieurs années. Le pilote noyau pour LVM 1 est inclus dans les noyaux 2.4, mais cela ne veut pas dire que votre noyau 2.4.x est à jour de la dernière version de LVM. Regardez dans le fichier README [ftp://ftp.sistina.com/pub/LVM/1.0/README] pour obtenir les dernières informations sur les versions de noyaux possédant le code le plus récent.

Chapitre 2. Qu'est-ce que la gestion par volumes logiques ?

La gestion par volumes logiques permet une gestion de plus haut niveau des problèmes de stockage sur disque par rapport à l'approche traditionnelle avec des disques et des partitions. Cela apporte à l'administrateur système une bien meilleure flexibilité pour allouer de l'espace aux applications et aux utilisateurs.

Les volumes de stockage créés au moyen du gestionnaire de volumes logiques peuvent être redimensionnés et déplacés à la demande, bien que cela puisse nécessiter une mise à jour des utilitaires des systèmes de fichiers.

Le gestionnaire de volumes logiques autorise aussi la gestion de volumes logiques par groupes d'utilisateurs, ce qui permet à l'administrateur système de gérer des groupes de volumes en utilisant un nommage compréhensible avec, par exemple, des noms comme « developpement » et « ventes » plutôt que les noms des disques physiques comme « sda » et « sdb ».

1. Pourquoi en voudrais-je ?

La gestion par volumes logiques est traditionnellement associée aux grosses installations contenant de nombreux disques mais elle est également appropriée pour les petites installations avec un ou deux disques.

2. Bénéfices de la gestion par volumes logiques sur un petit système

Le partitionnement du disque dur est l'une des décisions difficiles pour un nouvel utilisateur de Linux. La nécessité de devoir estimer précisément l'espace disque pour les fichiers systèmes et utilisateurs rend l'installation plus complexe que nécessaire et certains utilisateurs choisissent simplement de mettre toutes leurs données sur une seule grosse partition pour essayer de contourner ce problème.

Une fois que l'utilisateur (ou le programme d'installation) a déterminé l'espace nécessaire pour /home, /usr et /, il arrive fréquemment qu'une des partitions soit saturée alors qu'il reste encore plein d'espace libre sur une autre partition.

Avec la gestion par volumes logiques, le disque entier est alloué à un groupe de volumes et des volumes logiques sont créés pour contenir les systèmes de fichiers /, /usr et /home. Si, par exemple, le volume logique /home est plus tard saturé mais qu'il reste de l'espace libre dans /usr, alors il est possible de réduire /usr de quelques mégaoctets et de réallouer cet espace à /home.

Une autre solution est d'allouer un espace minimal pour chaque volume logique et de laisser une partie de l'espace disque non alloué. Ensuite, lorsque les partitions commencent à être remplies, elles peuvent être étendues suivant les besoins.

Par exemple : Joe achète un PC avec 8,4 Go d'espace disque et installe Linux avec le partitionnement suivant :

```
/boot    /dev/hda1     10 Mo
swap     /dev/hda2    256 Mo
/        /dev/hda3      2 Go
/home    /dev/hda4      6 Go
```

Il pense que cette répartition optimisera l'espace disponible pour tous ses fichiers MP3.

Quelque temps plus tard, Joe décide qu'il veut installer la dernière suite bureautique et le dernier environnement graphique disponible, mais il réalise que la partition racine n'est pas assez grande. Par contre, puisqu'il a archivé tous ses MP3 grâce à son nouveau graveur DVD, il lui reste plein d'espace libre dans /home.

Les différentes possibilités qui s'offrent à lui ne sont pas satisfaisantes :

1. Reformater le disque, modifier le partitionnement et tout réinstaller.
2. Acheter un nouveau disque et trouver un nouveau partitionnement qui demandera le moins de déplacements de données.
3. Utiliser un lien symbolique de / vers /home et installer le nouveau logiciel dans /home.

Avec LVM, tout cela devient beaucoup plus facile :

Jane achète un PC identique mais utilise LVM pour diviser son disque de manière similaire :

```
/boot     /dev/hda1         10 Mo
swap      /dev/vg00/swap   256 Mo
/         /dev/vg00/root     2 Go
/home     /dev/vg00/home     6 Go
```

Note

La partition de démarrage /boot n'est pas incluse dans le LV car les chargeurs de démarrage ne gèrent pas encore les volumes LVM. Il est possible que le démarrage sur une partition LVM fonctionne, mais vous risquez de vous retrouvez avec un système non amorçable.

Seuls les utilisateurs avancés peuvent envisager de passer leur partition racine en LVM

Pour passer la partition racine en LVM, une image initrd doit activer le LV racine au démarrage. Si le noyau est mis à jour sans créer l'image initrd nécessaire, le noyau ne pourra pas démarrer. Les nouvelles distributions incluent LVM dans leurs scripts mkinitrd ainsi que dans leurs images initrd, donc cela va devenir de moins en moins problématique.

Lorsque Jane rencontre le même problème, elle peut réduire la taille de /home d'un gigaoctet et rajouter cet espace dans la partition racine.

Supposons que Joe et Jane remplissent aussi la partition /home et décident d'ajouter un nouveau disque de 20 Go dans leur système.

Joe formate le disque entièrement dans une seule partition (/dev/hdb1) et l'utilise comme /home en y déplaçant les données de son /home existant. Il lui reste alors 6 Go inutilisés qu'il peut faire apparaître comme une extension de /home avec un lien symbolique, /home/joe/anciens-mp3 par exemple.

Jane ajoute simplement le nouveau disque dans son groupe de volumes existant et étend son /home pour inclure le nouveau disque. Ou bien, elle pourrait déplacer les données de /home de l'ancien vers le nouveau disque et ensuite étendre le volume racine existant pour couvrir tout l'ancien disque.

3. Avantages de la gestion par volumes logiques sur un gros système

3. Avantages de la gestion par volumes logiques sur un gros système

Les avantages de la gestion par volumes logiques sont plus évidents sur les gros systèmes avec de nombreux disques.

Gérer une importante ferme de disques prend beaucoup de temps et peut s'avérer particulièrement complexe si le système comporte de nombreux disques de tailles différentes. Gérer les problèmes d'espace, généralement conflictuels, entre les différents utilisateurs peut devenir un cauchemar.

Les différents groupes de volumes et volumes logiques, qui peuvent être agrandis autant que nécessaire, peuvent être alloués à des groupes d'utilisateurs. L'administrateur peut « garder en réserve » de l'espace jusqu'à ce qu'il soit requis. Cet espace peut alors être ajouté au groupe de volumes qui en a le plus besoin.

Quand de nouveaux disques sont ajoutés au système, il n'est plus nécessaire de déplacer les fichiers des utilisateurs pour optimiser le nouvel espace de stockage. Il suffit simplement d'ajouter le nouveau disque à un ou plusieurs groupes de volumes et d'étendre les volumes logiques suivant les besoins.

Il est aussi facile de se séparer des anciens disques en déplaçant les données sur des disques plus récents. Ceci peut être effectué à chaud, sans coupure de service pour l'utilisateur.

Chapitre 3. Anatomie de LVM

Le schéma suivant donne une vision générale des composants d'un système à base de LVM.

```
+-- Groupe de volumes (Volume Group) ------------+
|                                                |
|    +-----------------------------------------+ |
|  PV | PE |  PE | PE | PE | PE | PE | PE | PE | |
|    +-----------------------------------------+ |
|      .          .         .         .         |
|      .          .         .         .         |
|    +-----------------------------------------+ |
|  LV | LE |  LE | LE | LE | LE | LE | LE | LE | |
|    +-----------------------------------------+ |
|            .         .         .         .    |
|            .         .         .         .    |
|    +-----------------------------------------+ |
|  PV | PE |  PE | PE | PE | PE | PE | PE | PE | |
|    +-----------------------------------------+ |
|                                                |
+------------------------------------------------+
```

Voici une autre façon de visualiser ceci (grâce à la gentillesse d'Erik Bågfors [mailto:erik CHEZ bagfors POINT nu_NOSPAM] de la liste de diffusion linux-lvm) :

```
    hda1   hdc1      (PV dans des partitions ou disques entiers)
       \   /
        \ /
       diskvg        (VG)
       /  |  \
      /   |   \
  usrlv rootlv varlv (LV)
    |      |     |
 ext2  reiserfs  xfs (Systèmes de fichiers)
```

1. Groupe de volumes (Volume Group, VG)

Le groupe de volumes est l'abstraction de plus haut niveau utilisée dans LVM. C'est un ensemble de volumes logiques et physiques au sein d'une seule unité d'administration.

2. Volume physique (Physical Volume, PV)

Un volume physique est typiquement un disque dur, bien que cela puisse aussi être un périphérique qui « ressemble » à un disque dur (par exemple, un périphérique de RAID logiciel).

3. Volume logique (Logical Volume, LV)

C'est l'équivalent d'une partition de disque d'un système sans LVM. Le LV est visible comme un périphérique de bloc standard. En tant que tel, le LV peut contenir un système de fichiers. (Ex : `/home`.)

4. Extent physique (Physical Extent, PE)

Chaque volume physique est divisé en morceaux de données, appelés extents physiques. Ces extents ont une taille identique à celle des extents logiques du groupe de volumes.

5. Extent logique (Logical Extent, LE)

Chaque volume logique est divisé en morceaux de données, appelés extents logiques. La taille d'extents est la même pour tous les volumes logiques du groupe de volumes.

6. Mettre tout ensemble

Voici un exemple concret pour aider à la compréhension :

Prenons un groupe de volumes nommé VG1, avec une taille d'extent physique de 4 Mo. Nous mettons dans ce groupe de volumes deux partitions de disques, /dev/hda1 et /dev/hdb1. Ces partitions vont devenir les volumes physiques PV1 et PV2 (l'administrateur peut mettre des noms plus parlants). Les PV sont divisés en morceaux de 4 Mo, puisque c'est la taille des extents pour le groupe de volumes. Les disques sont de tailles différentes et nous avons donc 99 extents pour PV1 et 248 pour PV2. Nous pouvons maintenant créer un volume logique de taille entre 1 et 347 (248 + 99) extents. Quand le volume logique est créé, une relation est définie entre les extents logiques et physiques, c'est-à-dire que l'extent logique 1 peut correspondre à l'extent physique 51 de PV1 et les données écrites sur les quatre premiers mégaoctets du volume logique seront en fait écrites sur le 51ème extent de PV1.

7. Types de correspondances (linéaire/répartie)

L'administrateur peut utiliser différentes stratégies pour effectuer la relation entre les extents logiques et les extents physiques :

1. La **correspondance linéaire** permet d'attribuer une plage de PE à un LV dans l'ordre. Par exemple, les LE 1 à 99 correspondront à PV1 et les LE 100 à 347 à PV2.

2. La **correspondance répartie (striped)** permet de distribuer des morceaux d'extents logiques sur plusieurs volumes physiques, par exemple :

```
1er morceau de LE[1] -> PV1[1],

2me morceau de LE[1] -> PV2[1],

3me morceau de LE[1] -> PV3[1],

4me morceau de LE[1] -> PV1[2],
```

   et ainsi de suite. Dans certaines situations, cette stratégie peut améliorer les performances du volume logique.

   Défaut de LVM 1

   Avec LVM 1, les LV créés par répartition ne peuvent pas être étendus au-delà des PV où ils ont été créés initialement.

   Avec LVM 2, les LV répartis peuvent être étendus en concaténant un autre lot de périphériques à la fin du premier lot. Par conséquent, vous pourrez vous retrouver avec un LV composé de deux lots de PE répartis concaténés avec un lot linéaire composé de quatre lots de PE. Je vous ai déjà perdu ?

8. Les instantanés (snapshots)

Une des merveilleuses fonctionnalités apportées par LVM est l'instantané. Elle permet à l'administrateur de créer un nouveau périphérique qui est une copie exacte d'un volume logique figé à un moment donné. Typiquement, cela peut être utilisé lors de traitements par lots sur le volume logique, pour une sauvegarde par exemple, lorsque l'on ne veut pas arrêter une application qui est en train de modifier les données. Une fois que l'on a fini d'utiliser l'instantané, l'administrateur système peut tout simplement supprimer le périphérique. Cette fonctionnalité demande à ce que l'instantané soit effectuée lorsque le volume logique se trouve dans un mode consistant. La suite du document en donne quelques exemples.

Mise à jour de LVM 1 vers LVM 2

Assurez-vous de supprimer tous les instantanés avant la mise à jour de LVM 1 vers LVM 2. (Cf. Section 1, « FAQ LVM 2 ».)

De plus amples informations sur les instantanés sont disponibles dans la Section 4, « Faire une sauvegarde avec un instantané (snapshot) ».

Chapitre 4. Questions fréquemment posées

1. FAQ LVM 2

1.1.
J'ai LVM 1 installé et fonctionnel sur mon système. Comment faire pour passer à LVM 2 ?

Voici les instructions pour démarrer rapidement :)

1. Commencez par enlever tous les instantanés du système. Ils ne sont pas gérés par LVM 2 et empêcheront l'activation de leur volume d'origine au démarrage de LVM 2.
2. Assurez-vous de posséder un moyen de démarrage du système autre que par la partition de démarrage standard. Vous devez avoir les utilitaires de LVM 1, les outils systèmes standards (mount) et un noyau compatible LVM 1 au cas où il serait nécessaire de revenir en arrière pour faire des corrections.
3. Récupérez les sources des utilitaires de LVM 2 et le mappeur de périphériques. Compilez-les. Il faut installer la bibliothèque du mappeur de périphériques avec « make install » avant de compiler les utilitaires de LVM 2. Copiez aussi le script dm/scripts/devmap_mknod.sh dans /sbin. Je recommande d'installer seulement le binaire « lvm » pour l'instant pour garder l'accès aux utilitaires de LVM 1 au cas où. Si vous avez accès à des paquets de LVM 2 et du mappeur de périphériques, vous pouvez les installer à la place mais il faut noter qu'il vont probablement écraser les utilitaires de LVM 1.
4. Récupérez un noyau compatible avec le mappeur de périphériques, compilé en interne ou avec un module.
5. Recherchez les commandes d'activation de LVM dans les scripts de démarrage de la machine. Assurez-vous que le module du mappeur de périphériques est chargé à cet endroit (si vous utilisez le mappeur de périphériques en tant que module) et ajoutez ensuite la commande « /sbin/devmap_mknod.sh; lvm vgscan; lvm vgchange -ay ».
6. Installez le noyau compatible avec le mappeur de périphériques et redémarrez. Si tout se passe bien, vous devriez être en train d'utiliser LVM 2.

1.2.
J'ai des erreurs concernant `/dev/mapper/control` quand j'essaie d'utiliser les utilitaires LVM 2. Que se passe-t-il ?

La cause première de ce problème est de ne pas avoir lancé le script devmap_mknod.sh après le redémarrage avec un noyau utilisant le mappeur de périphériques. Ce script génère le nœud de contrôle pour le mappeur de périphériques.

Si vous n'avez pas le script devmap_mknod.sh, ne désespérez pas, il est assez simple de créer le fichier `/dev/mapper/control` par vous même :

1. Assurez-vous d'avoir chargé le module du mappeur de périphériques s'il n'est pas directement dans le noyau.
2. Tapez :

```
# cat /proc/misc | grep device-mapper | awk '{print $1}'
```

et notez le chiffre affiché. (S'il n'y a rien d'affiché, reportez-vous à l'étape 1.)

3. Tapez :

```
# mkdir /dev/mapper
```

Si vous obtenez une erreur comme quoi `/dev/mapper` existe déjà, vérifiez que c'est un répertoire et continuez.

4. Tapez :

```
# mknod /dev/mapper/control c 10 $nombre
```

$nombre est le chiffre affiché à l'étape 2.

Cela devrait être bon maintenant.

1.3.

Quelles sont les commandes et types de volumes logiques compatibles avec LVM 2 ?

Si vous utilisez le correctif de l'archive lvm2 pour les noyaux stables 2.4, toutes les fonctionnalités majeures de LVM1 seront compatibles avec les outils de LVM2 (il est toujours nécessaire de supprimer les instantanés avant la mise à jour de LVM 1 vers LVM 2.)

Si vous utilisez la version du mappeur de périphériques du noyau 2.6 de kernel.org, les commandes et types de LV suivants ne sont pas compatibles :

- pvmove
- snapshots

Vous pourrez trouver un début de compatibilité avec ces fonctionnalités dans les correctifs instables du mappeur de périphériques [http://people.sistina.com/~thornber/dm/] maintenus par Joe Thornber.

1.4.

Les formats de représentation sur le disque des groupes de volumes et volumes logiques sont-ils différents dans LVM 2 par rapport à LVM 1 ?

Oui. LVM 2 utilise le format de méta-données lvm 2. Ce format est beaucoup plus flexible que celui de LVM 1. Il supprime ou réduit la plupart des limitations de LVM 1.

1.5.

LVM 2 est-il compatible avec les VG et LV créés avec LVM 1 ?

Oui. LVM 2 est capable d'activer et de fonctionner avec les VG et LV créés avec LVM 1 à l'exception des instantanés qui doivent être supprimés avant la mise à jour. Les instantanés qui restent après la mise à jour devront être supprimés afin que l'on puisse activer leur volume d'origine avec LVM 2.

1.6.

Puis-je passer mes VG et LV qui proviennent de LVM 1 au format natif de LVM 2 ?

Oui. vgconvert permet de convertir un VG et tous les LV qu'il contient au nouveau format de méta-données lvm 2. Attention, il n'est pas toujours possible de revenir ensuite aux méta-données lvm 1.

1.7.

J'ai mis à jour mon système avec LVM 2, mais les utilitaires échouent constamment en indiquant qu'il ne reste plus de mémoire libre. Que se passe-t-il ?

Une des causes possibles de cela est le non positionnement par certaines versions de LVM 1 du champ UUID dans les structures de PV et VG alors qu'elles étaient supposées le faire. (L'utilisateur qui a signalé l'anomalie utilisait une Mandrake 9.2, mais d'autres distributions peuvent être impactées.) Les dernières versions des utilitaires de LVM 2 positionnent les UUID automatiquement s'ils manquent, il vous suffit donc de récupérer une version plus récente que la vôtre et le problème devrait être résolu. Dans le cas contraire, envoyez un message sur la liste de diffusion linux-lvm.

1.8.

Ma partition racine est sur un LV en LVM 1. Comment faire pour passer à LVM 2 ? Qu'est devenu lvmcreate_initrd ?

Passer à LVM 2 avec sa partition racine en LVM est un peu compliqué, mais pas impossible. Vous devez créer un noyau compatible avec le mappeur de périphériques et installer les utilitaires LVM 2. (Vous devriez faire une sauvegarde des utilitaires de LVM 1 ou trouver un disque de secours avec les utilitaires LVM au cas où vous en auriez besoin avant la fin de la migration.) Ensuite, vous devez trouver un script mkinitrd compatible avec votre distribution et de LVM 2.

Voici la liste des scripts mkinitrd que je connais et qui sont compatibles avec LVM 2, classés par distributions :

Scripts mkinitrd compatible LVM 2

Fedora
: Le programme mkinitrd [http://distro.ibiblio.org/pub/linux/distributions/fedora/linux/core/development/i386/Fedora/RPMS/mkinitrd-3.5.21-1.i386.rpm] de la dernière Fedora Core 2 gère LVM 2, mais repose sur un binaire lvm lié statiquement dans la dernière archive LVM 2.

  Les utilisateurs de Redhat 9 devraient pouvoir l'utiliser.

Debian
: Il existe une version non officielle ici [http://www.poochiereds.net/svn/lvm2/].

Générique
: Il en existe une version dans la distribution source de LVM 2 dans le répertoire `scripts/lvm2_createinitrd/`. Voir la documentation dans ce répertoire pour plus de détails.

1.9.

Comment se comporte LVM par rapport à un renommage soudain des disques physiques ?

Très bien : LVM identifie les PV par leur UUID et non par le nom du périphérique.

Chaque disque (PV) possède un UUID, qui est un identifiant unique dans le système. Il est utilisé par « vgscan » pour le reconnaître lorsqu'un nouveau disque est ajouté et change la numérotation du disque. Pour cela, la plupart des distributions lancent vgscan dans les scripts de démarrage pour détecter les ajouts de matériel. Si vous faites des ajouts à chaud, je pense que vous devrez faire cela manuellement. D'un autre coté, si votre VG est activé et utilisé, la renumérotation ne devrait pas l'affecter du tout. L'identifiant est uniquement utilisé lors de l'activation et le pire qu'il puisse arriver sans le vgscan est que l'activation échoue avec un message indiquant qu'il manque un PV.

Note

Des problèmes de fonctionnement ou l'enlèvement d'un disque en cours d'utilisation par LVM cause des problèmes avec l'utilisation en cours et les futures activations du VG concerné.

1.10.
J'essaie de remplir mon VG et vgdisplay/vgs indique que j'ai 1,87 Go de libre mais quand je fais un lvcreate vg -L1.87G, il indique « pas assez d'extents libres » (« insufficient free extents »). Que se passe-t-il ?

Le chiffre 1,87 Go est arrondi à deux décimales, donc il correspond à quelque chose comme 1,866 Go. C'est juste une présentation à l'écran du chiffre pour donner une idée générale de la taille du VG. Si vous voulez donner une taille exacte, vous devez utiliser les extents à la place de quelque multiple d'octets.

Dans le cas de vgdisplay, utilisez le chiffre donné par Free PE à la place.

```
Free  PE / Size          478 / 1.87 GB
                         ^^^
```

Vous devriez donc lancer la commande suivante :

```
# lvcreate vg -l478
```

Remarquez qu'à la place d'un « L » majuscule, il faut utiliser un « l » minuscule pour dire à LVM d'utiliser les extents plutôt que les octets.

Dans le cas de vgs, il est nécessaire de lui demander explicitement le nombre d'extents disponibles :

```
# vgs -o +vg_free_count,vg_extent_count
```

Cela demande à vgs de rajouter le nombre d'extents libres et leur nombre total à la fin de la liste produite par vgs. Utilisez le nombre d'extents libres de la même façon que pour le cas de vgdisplay.

2. FAQ LVM 1

2.1.
Quand y aura-t-il des infos ici ?

Quand les gens commenceront à envoyer des entrées de FAQ ;)

Chapitre 5. Obtenir LVM

La première chose à faire est de récupérer une copie de LVM.

- Télécharger une archive de LVM par FTP.
- Télécharger le code source en développement continu par CVS.

1. Télécharger le code source

- Mappeur de périphériques [ftp://sources.redhat.com/pub/dm/]
- LVM 2 [ftp://sources.redhat.com/pub/lvm2/]

  Assurez-vous de télécharger aussi les sources du mappeur de périphériques.
- LVM 1 [ftp://sources.redhat.com/pub/lvm/current/]

Note

Le correctif du noyau pour LVM 1 doit être généré avec les sources de LVM 1. Plus d'informations sur ceci peuvent être trouvées dans la Section 2, « Construire le module du noyau pour LVM 1 ».

2. Télécharger le code source de développement par CVS

Remarque : L'état du code dans l'entrepôt CVS évolue sauvagement. Il contient des erreurs, certaines peuvent faire planter LVM ou le noyau. Il peut aussi ne pas compiler du tout. Considérez-le comme du code de qualité alpha et que vous pourriez perdre des données... Vous êtes prévenu.

3. Avant de commencer

Pour suivre la progression des développements de LVM, abonnez-vous aux listes de diffusion de LVM, linux-lvm et la liste de commit appropriée. (Cf. Section 1, « Listes de diffusion ».)

Pour construire LVM à partir des sources CVS, plusieurs outils GNU sont **nécessaires** :

- le client CVS version 1.9 ou supérieure ;
- GCC 2.95.2 ;
- GNU make 3.79 ;
- autoconf version 2.13 ou supérieure.

4. Configuration initiale

Pour vous rendre la vie plus facile lors des mises à jour futures de l'arbre CVS, créez le fichier

`$HOME/.cvsrc` et mettez-y les lignes suivantes qui permettent de configurer le comportement par défaut des commandes CVS les plus utilisées. Faites-le avant de continuer plus loin.

```
diff -u -b -B
checkout -P
update -d -P
```

De même, si vous utilisez une connexion à bas débit (comme un modem RTC), vous voudrez probablement rajouter une ligne contenant `cvs -z5` dans ce fichier. Cela active un niveau de compression qui sera utilisé par toutes les commandes CVS.

5. Récupération du code source

- **Bibliothèque et utilitaires du mappeur de périphériques**

  La bibliothèque du mappeur de périphériques est nécessaire pour construire LVM 2.

  Vous devez vous authentifier lors de votre premier chargement :

  ```
  # cvs -d :pserver:cvs@sources.redhat.com:/cvs/dm login cvs
  ```

  Le mot de passe est « cvs ». La commande ne renvoie rien en cas de succès et renvoie un message d'erreur en cas d'échec. Seule la première connexion nécessite une authentification explicite. Toutes les commandes CVS suivantes lisent le mot de passe stocké dans le fichier `$HOME/.cvspass` pour l'authentification.

  La commande de « checkout » suivante permet de récupérer une copie du code :

  ```
  # cvs -d :pserver:cvs@sources.redhat.com:/cvs/dm checkout device-mapper
  ```

  Cela crée un nouveau répertoire `device-mapper` dans le répertoire courant. Il contient la version la plus à jour possible du code du mappeur de périphériques.

- **LVM 2**

  Vous devez vous authentifier lors de votre premier chargement :

  ```
  # cvs -d :pserver:cvs@sources.redhat.com:/cvs/lvm2 login cvs
  ```

  Le mot de passe est « cvs ». La commande ne renvoie rien en cas de succès et un message d'erreur en cas d'échec. Seule la première connexion nécessite une authentification explicite. Toutes les commandes CVS suivantes lisent le mot de passe stocké dans le fichier `$HOME/.cvspass` pour l'authentification.

  La commande de « checkout » suivante permet de récupérer une copie du code :

  ```
  # cvs -d :pserver:cvs@sources.redhat.com:/cvs/lvm2 checkout LVM2
  ```

  Cela crée un nouveau répertoire `LVM2` dans le répertoire courant. Il contient la version la plus à jour possible du code de LVM 2.

- **LVM 1**

  Vous devez vous authentifier lors de votre premier chargement :

```
# cvs -d :pserver:cvs@sources.redhat.com:/cvs/lvm login cvs
```

Le mot de passe est « cvs ». La commande ne renvoie rien en cas de succès et un message d'erreur en cas d'échec. Seule la première connexion nécessite une authentification explicite. Toutes les commandes CVS suivantes lisent le mot de passe stocké dans le fichier `$HOME/.cvspass` pour l'authentification.

La commande de « checkout » suivante permet de récupérer une copie du code :

```
# cvs -d :pserver:cvs@sources.redhat.com:/cvs/lvm checkout LVM
```

Cela crée un nouveau répertoire `LVM` dans le répertoire courant. Il contient la version la plus à jour possible du code de LVM 1.

Les commandes CVS fonctionnent depuis *n'importe où* dans l'arborescence des sources et de façon récursive. Par conséquent, si vous faites une mise à jour à partir du sous répertoire « tools », cela fonctionnera mais uniquement pour le répertoire « tools » et ses sous-répertoires. Les commandes qui suivent supposent que vous êtes à la racine de l'arborescence des sources.

6. Mise à jour du code

Les modifications du code sont assez fréquentes dans le dépôt CVS. Elles sont annoncées automatiquement par un envoi dans la liste lvm-commit.

Vous pouvez mettre à jour votre copie des sources pour correspondre au dépôt maître avec la commande update. Il n'est pas nécessaire de récupérer une nouvelle copie des sources. Il est beaucoup plus simple et rapide d'utiliser la commande update, car elle télécharge uniquement les modifications des fichiers modifiés depuis la dernière mise à jour. Elle permet aussi de fusionner automatiquement les modifications du dépôt CVS avec les changements que vous auriez pu effectuer en local. Pour cela, il vous suffit juste d'aller dans le répertoire à mettre à jour et de taper la commande suivante :

```
# cvs update
```

Si vous n'avez pas spécifié de balise lors du « checkout », cela mettra à jour vos sources par rapport à la dernière version de la branche principale des sources. Si vous aviez spécifié une balise de branche, cela mettra à jour suivant la dernière version de cette branche. Si vous aviez spécifié une balise de version, cela ne fera rien du tout.

7. Commencer un projet

Discutez de l'idée sur la liste de développement avant de commencer. Quelqu'un est peut-être déjà en train de travailler sur le même sujet ou peut avoir des bonnes idées pour le mener à bien.

8. Bidouiller le code

Ainsi, vous voulez corriger un bogue ? Développer une fonctionnalité de la liste TODO ? Développer une nouvelle fonctionnalité ? Bidouiller le code ne pourrait pas être plus simple. Utilisez votre copie des sources. Pas besoin de copier les fichiers en `.orig` ou autres. CVS garde la copie des originaux.

Quand votre code fonctionnera et aura été testé du mieux possible avec le matériel dont vous disposez, faites un correctif par rapport à la version *current* des sources sur CVS.

```
# cvs update
# cvs diff > patchfile
```

Envoyez le correctif à la liste linux-lvm ou dm-devel (Section 1, « Listes de diffusion ») avec une description des modifications et ajouts que vous avez effectués.

9. Conflits

Si quelqu'un a travaillé sur les mêmes fichiers que vous, il se peut qu'il y ait des modifications conflictuelles. Vous le saurez quand vous mettrez à jour vos sources.

```
# cvs update
RCS file: LVM/tools/pvcreate.c,v
retrieving revision 1.5
retrieving revision 1.6
Merging differences between 1.5 and 1.6 into pvcreate.c
rcsmerge: warning: conflicts during merge
cvs server: conflicts found in tools/pvcreate.c
C tools/pvcreate.c
```

Pas de panique ! Le fichier original, tel qu'il était avant la mise à jour, est conservé avec le nom `.#pvcreate.c.1.5`. Vous pouvez toujours le récupérer si les choses se passent mal. Le fichier nommé `pvcreate.c` contient désormais les **deux** versions, la vôtre et la nouvelle, des lignes conflictuelles. Il vous reste simplement à ouvrir le fichier et résoudre les conflits en effaçant la mauvaise version des lignes.

```
<<<<<<< pvcreate.c
   j++;
=======
   j--;
>>>>>>> 1.6
```

N'oubliez pas de supprimer les lignes avec les symboles « < », « = » et « > ».

Chapitre 6. Construire les modules du noyau

1. Construire le module noyau device-mapper

À COMPLÉTER.

2. Construire le module du noyau pour LVM 1

Pour utiliser LVM 1, il faut que vous compiliez le module LVM 1 du noyau (recommandé) ou, si vous le préférez, que vous reconstruisiez le noyau avec le code LVM 1 lié statiquement dedans.

Votre système Linux est probablement basé sur une des distributions populaires de Linux (Redhat, Debian...) avec laquelle le module LVM 1 est peut-être déjà fourni. Vérifiez la version des utilitaires que vous avez sur votre système en utilisant une des commandes de LVM avec l'option « -h ». Utilisez la commande **pvscan -h** si vous ne connaissez aucune des commandes. Si le numéro de version en tête du listing de l'aide est 1.0.8, **utilisez votre configuration actuelle** et sautez le reste de ce chapitre.

2.1. Fabriquer un correctif pour votre noyau

Afin de mettre à jour le noyau pour qu'il soit compatible avec LVM v1.0.8, vous devez effectuer les opérations suivantes :

1. Décompresser LVM 1.0.8 :

   ```
   # tar zxf lvm_1.0.8.tar.gz
   ```

2. Allez dans le répertoire racine de cette version :

   ```
   # cd LVM/1.0.8
   ```

3. Lancez le script de configuration :

   ```
   # ./configure
   ```

   Il faut ajouter l'option `--with-kernel_dir` à configure si les sources du noyau ne sont pas dans `/usr/src/linux`. (Lancez **./configure --help** pour visualiser toutes les options disponibles.)

4. Allez dans le répertoire PATCHES :

   ```
   # cd PATCHES
   ```

5. Lancez « make » :

   ```
   # make
   ```

Vous devriez maintenant avoir un correctif appelé `lvm-1.0.8-$KERNELVERSION.patch` dans le répertoire PATCHES. C'est le correctif du noyau pour LVM qui sera utilisé dans la suite de ce guide pratique.

6. Patchez le noyau :

```
# cd /usr/src/linux ; patch -pX < /répertoire/lvm-1.0.8-$KERNELVERSION.patch
```

2.2. Construire le module LVM pour Linux 2.2.17+

Un noyau de la série 2.2 doit être mis à jour avant de commencer la compilation. Allez voir ailleurs les instructions pour apprendre à appliquer des correctifs au noyau.

Correctifs :

1. **Correctif rawio**

   Le correctif raw_io de Stephen Tweedie est disponible à http://www.kernel.org/pub/linux/kernel/people/sct/raw-io

2. **Correctif lvm**

   Ce correctif doit être construit à partir du sous-répertoire PATCHES de la distribution de LVM 1. Vous trouverez plus d'informations dans la Section 2.1, « Fabriquer un correctif pour votre noyau ».

Une fois que les correctifs ont été correctement appliqués, il faut vous assurer que le module soit bien compilé. Les pilotes LVM 1 se trouvent dans la section « devices » de la configuration du noyau. Vous devriez aussi compiler les informations de /proc pour LVM.

Compilez les modules noyau comme d'habitude.

2.3. Compiler les modules LVM pour Linux 2.4

Le noyau 2.4 contient déjà LVM, mais vous devriez vérifier la présence de mises à jour sur le site web de LVM (c'est-à-dire que les noyaux 2.4.9 et inférieurs doivent utiliser le plus récent correctif LVM 1). En configurant le noyau, cherchez LVM 1 dans **Multi-device support (RAID and LVM)**. LVM 1 peut être compilé dans le noyau ou en tant que module. Compilez votre noyau et les modules et installez-les de la manière habituelle. Si vous décidez de compiler LVM en tant que module, il s'appellera `lvm-mod.o`.

Si vous voulez faire des instantanés avec ReiserFS, faites attention à appliquer le correctif `linux-2.4.x-VFS-lock` (il y en a une copie dans le répertoire `LVM/1.0.8/PATCHES`).

2.4. Vérifier le système de fichiers proc

Si le noyau est compilé avec le système de fichiers /proc (probablement), vous pouvez vérifier la présence de LVM par l'existence du répertoire /proc/lvm. S'il n'existe pas, il se peut que vous ayez à charger le module par la commande :

```
# modprobe lvm-mod
```

Si `/proc/lvm` n'existe toujours pas, vérifiez soigneusement votre configuration noyau.

2.4. Vérifier le système de fichiers proc

Quand LVM est actif, il existe des entrées dans `/proc/lvm` pour tous les volumes physiques, groupes de volumes et volumes logiques. De plus, le « fichier » appelé `/proc/lvm/global` donne le statut de LVM et montre la version du noyau LVM en cours d'utilisation.

Chapitre 7. Scripts de démarrage pour LVM 1

Les scripts de démarrage ne sont pas fournis par la distribution de LVM, bien qu'ils soient simples à faire soi-même.

Le démarrage de LVM demande juste ces deux commandes :

```
# vgscan
# vgchange -ay
```

Et l'arrêt une seule :

```
# vgchange -an
```

Suivez les instructions ci-dessous suivant la distribution de Linux que vous utilisez.

1. Caldera

Il est nécessaire d'ouvrir le fichier `/etc/rc.d/rc.boot`. Insérez les lignes de commandes vgscan et vgchange juste avant les lignes qui disent « Mounting local filesystems ».

Vous pouvez aussi modifier le fichier `/etc/rc.d/init.d/halt` pour désactiver les groupes de volume à l'arrêt de la machine. Insérer la commande

```
vgchange -an
```

près de la fin du fichier, juste après le démontage des systèmes de fichiers ou leur montage en lecture seule et avant le commentaire disant « Now halt or reboot ».

2. Debian

Si vous téléchargez le paquet Debian lvm-common (pour LVM 1) ou lvm2, un script d'initialisation devrait être installé pour vous.

Si vous installez LVM à partir des sources, vous devez créer votre propre fichier d'initialisation :

- Créez le script de démarrage `/etc/init.d/lvm` contenant les lignes suivantes :

```
#!/bin/sh

case "$1" in
  start)
    /sbin/vgscan
    /sbin/vgchange -ay
    ;;
  stop)
    /sbin/vgchange -an
    ;;
  restart|force-reload)
    ;;
esac
```

```
exit 0
```

- Ensuite, exécutez les commandes :

  ```
  # chmod 0755 /etc/init.d/lvm
  # update-rc.d lvm start 26 S . stop 82 1 .
  ```

  Remarquez les points dans la dernière commande.

3. Mandrake

Aucune modification des scripts d'initialisation ne devrait être nécessaire pour les versions actuelles de Mandrake.

4. Redhat

Pour la Redhat 7.0 et supérieure, vous ne devriez pas avoir besoin de modifier de script d'initialisation pour démarrer LVM s'il est inclus dans le noyau. Si LVM est compilé en tant que module, il peut être nécessaire de modifier `/etc/rc.d/rc.sysinit` pour charger le module en y ajoutant la ligne « modprobe lvm-mod » avant la partie disant :

```
# LVM initialization, take 2 (it could be on top of RAID)
if [ -e /proc/lvm -a -x /sbin/vgchange -a -f /etc/lvmtab ]; then
   action $"Setting up Logical Volume Management:" /sbin/vgscan &&
   /sbin/vgchange -a y
fi
```

Note

Cette portion de script est pour la RedHat 7.3. Les autres versions peuvent apparaître légèrement différentes.

Pour les versions de RedHat antérieures à la 7.0, il est nécessaire de modifier le fichier `/etc/rc.d/rc.sysinit`. Regardez la ligne qui dit « Mount all other filesystems » et insérez les commandes vgscan et vgchange juste avant celle-ci. Assurez-vous que votre système de fichiers racine est monté en lecture/écriture avant d'utiliser les commandes LVM.

Vous voudrez peut-être aussi modifier le fichier `/etc/rc.d/init.d/halt` pour désactiver les groupes de volumes à l'arrêt de la machine. Insérez la commande :

```
vgchange -an
```

vers la fin du fichier, juste après le remontage des systèmes de fichiers en lecture seule et avant le commentaire qui dit « Now halt or reboot ».

5. Slackware

La Slackware 8.1 ne demande aucune mise à jour des scripts de démarrage pour faire fonctionner LVM.

Pour les versions précédentes de Slackware 8.1, vous devrez appliquer le correctif suivant à `/etc/rc.d/rc.S`.

```
cd /etc/rc.d
cp -a rc.S rc.S.old
patch -p0 < rc.S.diff
```

(Le cp sert à faire une sauvegarde au cas où).

```
----- snip snip file: rc.S.diff---------------
--- rc.S.or        Tue Jul 17 18:11:20 2001
+++ rc.S        Tue Jul 17 17:57:36 2001
@@ -4,6 +4,7 @@
#
# Mostly written by:  Patrick J. Volkerding, <volkerdi@slackware.com>
#
+# Added LVM support <tgs@iafrica.com>

PATH=/sbin:/usr/sbin:/bin:/usr/bin

@@ -28,19 +29,21 @@
READWRITE=yes
fi

+
# Check the integrity of all filesystems
if [ ! READWRITE = yes ]; then
-  /sbin/fsck -A -a
+  /sbin/fsck -a /
+  # Check only the root fs first, but no others
# If there was a failure, drop into single-user mode.
if [ ? -gt 1 ] ; then
echo
echo
-    echo "*****************************************************"
-    echo "*** An error occurred during the file system check. ***"
-    echo "*** You will now be given a chance to log into the  ***"
-    echo "*** system in single-user mode to fix the problem.  ***"
-    echo "*** Running 'e2fsck -v -y <partition>' might help.  ***"
-    echo "*****************************************************"
+    echo "**********************************************************"
+    echo "*** An error occurred during the root file system check. ***"
+    echo "*** You will now be given a chance to log into the       ***"
+    echo "*** system in single-user mode to fix the problem.       ***"
+    echo "*** Running 'e2fsck -v -y <partition>' might help.       ***"
+    echo "**********************************************************"
echo
echo "Once you exit the single-user shell, the system will reboot."
echo
@@ -82,6 +85,44 @@
echo -n "get into your machine and start looking for the problem. "
read junk;
fi
+  # okay / fs is clean, and mounted as rw
+  # This was an addition, limits vgscan to /proc thus
+  # speeding up the scan immensely.
+  /sbin/mount /proc
+
+  # Initialize Logical Volume Manager
+  /sbin/vgscan
+  /sbin/vgchange -ay
+
+  /sbin/fsck -A -a -R
+  #Check all the other filesystem, including the LVM's, excluding /
+
+  # If there was a failure, drop into single-user mode.
+  if [ ? -gt 1 ] ; then
+    echo
+    echo
```

```
+    echo "*****************************************************"
+    echo "*** An error occurred during the file system check. ***"
+    echo "*** You will now be given a chance to log into the  ***"
+    echo "*** system in single-user mode to fix the problem.  ***"
+    echo "*** Running 'e2fsck -v -y <partition>' might help.  ***"
+    echo "*** The root filesystem is ok and mounted readwrite ***"
+    echo "*****************************************************"
+    echo
+    echo "Once you exit the single-user shell, the system will reboot."
+    echo
+
+    PS1="(Repair filesystem) #"; export PS1
+    sulogin
+
+    echo "Unmounting file systems."
+    umount -a -r
+    mount -n -o remount,ro /
+    echo "Rebooting system."
+    sleep 2
+    reboot
+  fi
+
else
echo "Testing filesystem status: read-write filesystem"
if cat /etc/fstab | grep ' / ' | grep umsdos 1> /dev/null 2> /dev/null ;
then
@@ -111,14 +152,16 @@
echo -n "Press ENTER to continue. "
read junk;
fi
+
fi

+
# remove /etc/mtab* so that mount will create it with a root entry
/bin/rm -f /etc/mtab* /etc/nologin /etc/shutdownpid

# mount file systems in fstab (and create an entry for /)
# but not NFS or SMB because TCP/IP is not yet configured
-/sbin/mount -a -v -t nonfs,nosmbfs
+/sbin/mount -a -v -t nonfs,nosmbfs,proc

# Clean up temporary files on the /var volume:
/bin/rm -f /var/run/utmp /var/run/*.pid /var/log/setup/tmp/*
--snip snip snip end of file---------------
```

6. SuSE

Aucun changement ne devrait être nécessaire à partir de la 6.4 puisque LVM y est inclus.

Chapitre 8. Scripts de démarrage pour LVM 2

À COMPLÉTER.

Chapitre 9. Compiler LVM à partir des sources

1. Compiler la bibliothèque et les utilitaires LVM

Allez dans le répertoire de LVM et faites un **./configure** suivi d'un **make**. Cela créera toutes les bibliothèques et programmes.

Si vous en avez besoin, vous pouvez modifier des options avec la commande de configuration. Faites un **./configure --help** pour voir les options disponibles. La plupart du temps, ceci sera inutile.

Il ne devrait pas y avoir d'erreur dans le processus de compilation. S'il y en a, reportez-vous au chapitre Signaler des erreurs et des bogues pour les signaler.

Vous êtes aussi les bienvenus si vous les corrigez et que vous nous envoyez les correctifs. Les correctifs sont habituellement à envoyer à la liste linux-lvm.

2. Installer la bibliothèque LVM et les utilitaires

Une fois que les sources compilent correctement, faites simplement un **make install** pour installer la bibliothèque LVM et les utilitaires sur votre système.

3. Supprimer la bibliothèque LVM et les utilitaires

Pour enlever la bibliothèque et les utilitaires que vous venez d'installer, faites **make remove**. Vous devez avoir les sources originelles utilisées pour l'installation de LVM pour que cela fonctionne.

Chapitre 10. Migrer d'une version précédente de LVM vers LVM 1.0.8

Migrer d'une version précédente de LVM vers LVM 1.0.8 devrait être très simple. Nous avons trouvé une méthode pour lire les méta-données des PV version 1 (LVM 0.9.1 Beta7 et précédents) aussi bien que les méta-données des PV version 2 (LVM 0.9.1 Beta8 et LVM 1.0).

Attention : Les nouveaux PV initialisés avec LVM 1.0.8 utilisent la structure de disque des PV version 1. Cela signifie que LVM 0.9.1 Beta8 et LVM 1.0 ne peuvent pas lire ou utiliser les PV créés avec la version 1.0.8.

1. Migrer vers LVM 1.0.8 avec une partition racine non LVM

Les étapes de la migration sont très simples, mais il est néanmoins recommandé de faire une sauvegarde des données avant. Vous êtes prévenus.

1. **Compiler le noyau LVM et les modules**

   Suivez les étapes expliquées dans les Chapitre 5, *Obtenir LVM* et Section 2, « Construire le module du noyau pour LVM 1 » pour connaître les instructions pour obtenir et compiler les éléments nécessaires de LVM pour le noyau Linux.

2. **Compiler les utilitaires LVM**

   Suivez les étapes du Chapitre 9, *Compiler LVM à partir des sources* pour compiler et installer les utilitaires de LVM.

3. **Configurer les scripts de démarrage**

   Assurez-vous d'avoir une configuration correcte des scripts de démarrage comme dans le Chapitre 7, *Scripts de démarrage pour LVM 1*.

4. **Démarrer avec le nouveau noyau**

   Assurez-vous que le chargeur d'amorçage est configuré pour charger le nouveau noyau avec LVM et, si vous utilisez des modules, que la commande **insmod lvm-mod** soit dans le script de démarrage ou que les lignes suivantes soient ajoutées dans `/etc/modules.conf` (anciennement appelé `/etc/conf.modules`) pour permettre à modprobe de charger le module LVM (n'oubliez pas d'activer kmod) :

   ```
   alias block-major-58      lvm-mod
   alias char-major-109      lvm-mod
   ```

   Redémarrez la machine et appréciez.

2. Migrer vers LVM 1.0.8 avec une partition racine en LVM et initrd

La procédure est relativement simple si elle est suivie avec attention. Il est recommandé de faire une bonne sauvegarde avec une disquette de démarrage de secours, juste au cas où.

L'utilisation « normale » de LVM sur le système de fichiers racine est d'avoir une seule partition non

LVM appelée `/boot` qui contient le noyau et une image disque RAM d'initialisation, nécessaire pour démarrer le système. Le système que j'ai mis à jour était comme suit :

```
# df
Sys. de fich.         Blocs-1k    Occupé    Dispo. Cap% Monté sur
/dev/rootvg/root        253871     93384    147380  39% /
/dev/hda1                17534     12944      3685  78% /boot
/dev/rootvg/home       4128448      4568   3914168   0% /home
/dev/rootvg/usr        1032088    332716    646944  34% /usr
/dev/rootvg/var         253871     31760    209004  13% /var
```

`/boot` contient l'ancien noyau et l'image disque RAM d'initialisation ainsi que les fichiers d'amorçage de Lilo et les lignes suivantes dans `/etc/lilo.conf` :

```
# ls /boot
System.map                 lost+found               vmlinux-2.2.16lvm
map                        module-info              boot.0300
boot.b                     os2_d.b                  chain.b
initrd.gz
# tail /etc/lilo.conf
image=/boot/vmlinux-2.2.16lvm
        label=lvm08
        read-only
        root=/dev/rootvg/root
        initrd=/boot/initrd.gz
        append="ramdisk_size=8192"
```

1. **Compiler le noyau LVM et les modules**

   Suivez les instructions détaillées dans les Chapitre 5, *Obtenir LVM* et Section 2, « Construire le module du noyau pour LVM 1 » pour obtenir et compiler les composants noyau nécessaires pour LVM.

2. **Compiler les utilitaires LVM**

   Suivez les instructions détaillées dans le Section 2, « Construire le module du noyau pour LVM 1 » pour compiler et installer les utilitaires nécessaires pour LVM.

   Installer les nouveaux utilitaires. Une fois cela effectué, vous ne pourrez plus faire de manipulations sur LVM car les utilitaires ne sont pas compatibles avec le noyau démarré actuellement.

3. **Renommer le fichier initrd.gz existant**

   Cela permet qu'il ne soit pas écrasé par le nouveau fichier.

   ```
   # mv /boot/initrd.gz /boot/initrd08.gz
   ```

4. **Modifier `/etc/lilo.conf`**

   Faites pointer l'entrée de démarrage existante vers le fichier renommé. Vous devrez redémarrer en utilisant cela si quelque chose se passe mal. L'entrée modifiée devrait ressembler à cela :

   ```
   image=/boot/vmlinux-2.2.16lvm
           label=lvm08
           read-only
           root=/dev/rootvg/root
           initrd=/boot/initrd08.gz
           append="ramdisk_size=8192"
   ```

5. **Lancer lvmcreate_initrd pour créer une nouvelle image disque RAM d'initialisation**

   ```
   # lvmcreate_initrd 2.4.9
   ```

   N'oubliez pas de spécifier la nouvelle version du noyau afin qu'il prenne les bons modules.

6. **Ajouter une nouvelle entrée dans /etc/lilo.conf**

   Cette nouvelle entrée sert à démarrer avec le nouveau noyau et le nouvel initrd.

   ```
   image=/boot/vmlinux-2.4.9lvm
   label=lvm10
   read-only
   root=/dev/rootvg/root
   initrd=/boot/initrd.gz
   append="ramdisk_size=8192"
   ```

7. **Relancer lilo**

   Cela installe le nouveau bloc d'amorçage

   ```
   # /sbin/lilo
   ```

8. **Redémarrer la machine**

   Lorsque vous avez l'écran de sélection de Lilo, choisissez la nouvelle entrée (lvm10 dans cet exemple) et le système devrait démarrer avec un Linux utilisant la nouvelle version de LVM.

   Si le nouveau noyau ne démarre pas, alors démarrez avec l'ancien et essayez de corriger le problème. Cela peut provenir du noyau qui n'aurait pas tous les pilotes de périphériques correctement compilés ou bien qu'ils ne sont pas disponibles dans initrd. Souvenez-vous que tous les pilotes (excepté pour LVM) qui sont nécessaires pour accéder au périphérique racine doivent être compilés en dur dans le noyau et non en tant que modules.

   Si vous avez besoin d'effectuer des manipulations avec LVM lorsque vous avez redémarré sur l'ancienne version, il vous faut juste recompiler les anciens utilitaires et les installer avec :

   ```
   # make install
   ```

   Si vous faites cela, n'oubliez pas de réinstaller la nouvelle version lorsque vous redémarrerez avec le nouveau LVM.

Une fois satisfait de la nouvelle installation, pensez à modifier l'entrée « default= » dans le fichier lilo.conf pour mettre par défaut le nouveau noyau.

Chapitre 11. Tâches courantes

Ce chapitre détaille quelques opérations courantes sur un système LVM. *Cela ne remplace pas la lecture des pages de man.*

1. Initialiser des disques ou des partitions de disques

Avant de pouvoir utiliser un disque ou une partition comme volume physique, il faut l'initialiser :

Pour un disque entier :

- Lancez pvcreate sur le disque :

  ```
  # pvcreate /dev/hdb
  ```

  Cela crée un descripteur de groupe de volumes au début du disque.

- Si vous obtenez une erreur indiquant que LVM ne peut pas initialiser un disque qui possède déjà une table de partition, vérifiez tout d'abord que vous êtes en train d'utiliser le bon disque. Si vous en êtes vraiment sûr, effectuez les commandes suivantes :

  DANGEREUX

  Les commandes suivantes vont détruire la table des partitions sur le disque où elles sont lancées. Soyez vraiment sûr que c'est le bon disque.

  ```
  # dd if=/dev/zero of=/dev/nomdudisque bs=1k count=1
  # blockdev --rereadpt /dev/nomdudisque
  ```

Pour les partitions :

- Quand vous utilisez LVM 1 sur des PC avec des partitions DOS, mettez le type de la partition à 0x8e avec fdisk ou un programme similaire. Ceci n'est pas nécessaire avec un système PPC ou avec LVM 2.

- Lancez pvcreate sur la partition :

  ```
  # pvcreate /dev/hdb1
  ```

  Cela crée un descripteur de groupe de volumes au début de la partition /dev/hdb1.

2. Créer un groupe de volumes

Utilisez le programme **vgcreate** :

```
# vgcreate mon_groupe_de_volumes /dev/hda1 /dev/hdb1
```

Remarque : Si vous utilisez devfs, il est primordial d'utiliser le nom complet du périphérique devfs plutôt que le lien symbolique dans `/dev`. Par conséquent la commande précédente devient :

```
# vgcreate mon_groupe_de_volumes \\
    /dev/ide/host0/bus0/target0/lun0/part1 \\
    /dev/ide/host0/bus0/target1/lun0/part1
```

LVM 2 n'a pas cette restriction.

Vous pouvez aussi préciser la taille des extents avec l'option « -s » de cette commande si la valeur par défaut de 32 Mo ne vous convient pas. De plus, vous pouvez mettre des limites sur le nombre de volumes physiques et logiques que le volume peut contenir.

3. Activer un groupe de volumes

Après un redémarrage ou la commande **vgchange -an**, les VG et LV ne sont plus accessibles. Pour réactiver le groupe de volumes, exécutez :

```
# vgchange -a y mon_groupe_de_volumes
```

4. Enlever un groupe de volumes

Assurez-vous qu'aucun volume logique n'est présent dans le groupe de volumes, voir les sections suivantes pour savoir comment faire.

Désactivez le groupe de volumes :

```
# vgchange -a n mon_groupe_de_volumes
```

Maintenant, vous pouvez supprimer le groupe de volumes :

```
# vgremove mon_groupe_de_volumes
```

5. Ajouter un volume physique à un groupe de volumes

Utilisez « vgextend » pour ajouter un volume physique déjà initialisé à un groupe de volumes existant.

```
# vgextend mon_groupe_de_volumes /dev/hdc1
                                 ^^^^^^^^^
                                 nouveau volume physique
```

6. Supprimer un volume physique d'un groupe de volumes

La commande « pvdisplay » permet de s'assurer que le volume physique n'est utilisé par aucun volume logique :

```
# 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-MOkn-76iP-iftz-OVSen7
```

Si le volume physique est encore utilisé, il faut migrer les données vers un autre volume physique.

Utilisez ensuite « vgreduce » pour enlever le volume physique :

```
# vgreduce mon_groupe_de_volumes /dev/hda1
```

7. Créer un volume logique

Pour créer un LV « testlv » linéaire de 1 500 Mo et son périphérique spécial « /dev/testvg/testlv » :

```
# lvcreate -L1500 -ntestlv testvg
```

Pour créer un volume logique de 100 LE avec 2 blocs répartis et une taille de bloc de 4 Ko :

```
# lvcreate -i2 -I4 -l100 -nunautretestlv testvg
```

Pour créer un LV qui utilise tout le VG, utilisez vgdisplay pour trouver la valeur de « Total PE », puis utilisez-la avec lvcreate.

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

Cela créera un LV appelé **monlv** qui remplira la totalité du VG **testvg**.

Si vous désirez allouer le volume logique sur un volume physique précis, il faut spécifier le nom du ou des PV à la fin de la commande lvcreate.

```
# lvcreate -L 1500 -ntestlv testvg /dev/sdg
```

8. Supprimer un volume logique

Un volume logique doit être démonté avant d'être supprimé :

```
# umount /dev/monvg/homevol
# lvremove /dev/monvg/homevol
lvremove -- do you really want to remove "/dev/monvg/homevol"? [y/n]: y
lvremove -- doing automatic backup of volume group "monvg"
lvremove -- logical volume "/dev/monvg/homevol" successfully removed
```

9. Étendre un volume logique

Pour étendre un volume logique, il suffit de dire à lvextend de combien vous voulez augmenter la taille. Vous pouvez spécifier la quantité d'espace à ajouter ou bien la taille finale du volume logique :

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

étend `/dev/monvg/homevol` jusqu'à 12 Go.

```
# lvextend -L+1G /dev/monvg/homevol
lvextend -- extending logical volume "/dev/monvg/homevol" to 13 GB
lvextend -- doing automatic backup of volume group "monvg"
lvextend -- logical volume "/dev/monvg/homevol" successfully extended
```

ajoute 1 Go à `/dev/monvg/homevol`.

Une fois le volume logique étendu, il est nécessaire d'augmenter la taille du système de fichiers à la taille correspondante. La procédure à suivre dépend du type de système de fichiers utilisé.

Par défaut, la plupart des utilitaires de redimensionnement de systèmes de fichiers vont augmenter leur taille suivant la taille du volume logique correspondant. Vous n'avez donc pas à vous occuper de spécifier la même taille pour les deux commandes.

1. **ext2/ext3**

   À moins que vous n'ayez un noyau avec le correctif ext2online, il est nécessaire de démonter le système de fichiers avant le redimensionnement. (Il semble que le patch de redimensionnement à chaud soit dangereux, donc à utiliser à vos risques et périls.)

   ```
   # umount /dev/monvg/homevol
   # resize2fs /dev/monvg/homevol
   # mount /dev/monvg/homevol /home
   ```

   Si vous n'avez pas e2fsprogs 1.19 ou supérieur, la commande ext2resize est disponible sur ext2resize.sourceforge.net [http://ext2resize.sourceforge.net]. Utilisez-la ensuite :

   ```
   # umount /dev/monvg/homevol
   # resize2fs /dev/monvg/homevol
   # mount /dev/monvg/homevol /home
   ```

   Pour ext2, il y a une solution plus simple : LVM 1 fournit un utilitaire appelé e2fsadm qui effectue le lvextend et le resize2fs pour vous (et peut aussi faire la réduction, voir le chapitre suivant).

   Inconvénient de LVM 2

   Il n'y a pas actuellement d'équivalent à e2fsadm pour LVM 2 et la version de e2fsadm de LVM 1 ne fonctionne pas avec LVM 2.

   La commande unique devient :

```
# e2fsadm -L+1G /dev/monvg/homevol
```

qui est équivalente à :

```
# lvextend -L+1G /dev/monvg/homevol
# resize2fs /dev/monvg/homevol
```

Remarque

Il est toujours nécessaire de démonter le système de fichiers avant d'utiliser e2fsadm.

2. **reiserfs**

   Le système de fichiers Reiserfs peut être redimensionné monté ou démonté, comme vous le préférez :

   - en fonctionnement :

     ```
     # resize_reiserfs -f /dev/monvg/homevol
     ```

   - hors fonctionnement :

     ```
     # umount /dev/monvg/homevol
     # resize_reiserfs /dev/monvg/homevol
     # mount -treiserfs /dev/monvg/homevol /home
     ```

3. **xfs**

   Les systèmes de fichiers XFS doivent être montés pour être redimensionnés et le point de montage doit être donné à la place du nom de périphérique.

   ```
   # xfs_growfs /home
   ```

4. **jfs**

   Comme XFS, le système de fichiers JFS doit être monté pour être redimensionné et le point de montage doit être spécifié plutôt que le nom du périphérique. Il vous faut au minimum la version 1.0.21 de jfs-utils pour cela.

   ```
   # mount -o remount,resize /home
   ```

Bogue connu du noyau

Certaines versions du noyau ont des problèmes avec cette syntaxe (2.6.0 en fait partie). Dans ce cas, vous devez explicitement spécifier la nouvelle taille du système de fichiers en blocs. Attention aux erreurs car vous *devez* connaître la taille de blocs de votre système de fichiers et vous en servir pour calculer la nouvelle taille.

Exemple : Si vous devez redimensionner un système de fichiers JFS à 4 Go, écrivez :

```
# mount -o remount,resize=1048576 /home
```

10. Réduire un volume logique

Les volumes logiques peuvent être réduits de la même façon qu'ils peuvent être agrandis. Cependant, il est *très* important de se souvenir qu'il faut réduire la taille du système de fichiers ou de toute autre chose résidant sur le volume avant de le réduire lui-même. Autrement, vous risquez de perdre des données.

1. **ext2**

   Si vous utilisez LVM 1 avec le système de fichiers ext2, vous pouvez utiliser e2fsadm comme mentionné précédemment pour effectuer les deux commandes de réduction du système de fichiers et de réduction du volume :

   ```
   # umount /home
   # e2fsadm -L-1G /dev/monvg/homevol
   # mount /home
   ```

   ### Inconvénient de LVM 2

   Actuellement, il n'y a pas d'équivalent à e2fsadm pour LVM 2 et la version de e2fsadm de LVM 1 ne fonctionne pas avec LVM 2.

   Si vous préférez faire cela manuellement, vous devez connaître la nouvelle taille du volume en blocs et utiliser les commandes suivantes :

   ```
   # umount /home
   # resize2fs /dev/monvg/homevol 524288
   # lvreduce -L-1G /dev/monvg/homevol
   # mount /home
   ```

2. **reiserfs**

   Il semble que Reiserfs préfère être démonté avant d'être réduit :

   ```
   # umount /home
   # resize_reiserfs -s-1G /dev/monvg/homevol
   # lvreduce -L-1G /dev/monvg/homevol
   # mount -treiserfs /dev/monvg/homevol /home
   ```

3. **xfs**

   Il n'y a pas moyen de réduire un système de fichiers XFS.

4. **jfs**

   Il n'y a pas moyen de réduire un système de fichiers JFS.

11. Migrer des données hors d'un volume physique

Pour mettre un disque hors service, il faut au préalable déplacer tous ses extents physiques sur d'autres disques du groupe de volumes. Il doit y avoir assez d'extents physiques libres sur les PV qui restent pour accueillir les extents provenant du disque à enlever. Pour plus de détails, voir la Section 5, « Enlever un disque ».

Chapitre 12. Partitionnement de disques

1. Plusieurs partitions sur le même disque

Le LVM permet de créer des PV (volumes physiques) sur presque tous les périphériques par blocs. Par exemple, les commandes suivantes sont valides et vont très bien fonctionner dans un environnement LVM :

```
# pvcreate /dev/sda1
# pvcreate /dev/sdf
# pvcreate /dev/hda8
# pvcreate /dev/hda6
# pvcreate /dev/md1
```

Dans un environnement de production « normal », il est recommandé de n'avoir qu'un seul PV par disque réel pour les raisons suivantes :

- Facilité d'administration.

  Il est plus simple de gérer le matériel du système si chaque disque réel n'apparaît qu'en un seul endroit. C'est particulièrement vrai en cas de problème sur le disque.

- Éviter les problèmes de performances avec la répartition.

  LVM ne sait pas dire si deux PV sont sur le même disque physique. Donc si vous créez des LV répartis, les blocs peuvent très bien être sur différentes partitions du même disque, ce qui entraîne une **diminution** des performances, plutôt qu'une augmentation.

Cependant, il peut être souhaitable de le faire pour certaines raisons :

- La migration d'un système existant vers LVM.

  Sur un système avec peu de disques, il peut être nécessaire de déplacer des données entre les partitions pour effectuer la conversion (cf. la Section 8, « Convertir le système de fichiers racine en LVM 1 »).

- Séparer un gros disque dans plusieurs groupes de volumes.

  Si vous avez un très gros disque et que vous voulez plus d'un groupe de volumes pour des raisons d'administration, alors il est nécessaire de faire plusieurs partitions sur le disque.

Si vous avez un disque avec plus d'une partition et que ces partitions sont dans le même groupe de volumes, faites attention à bien spécifier les partitions qui doivent être incluses dans le volume logique lorsque vous créez des volumes répartis.

Il est recommandé d'utiliser une seule partition qui couvre l'ensemble du disque physique. Cela évite des accidents avec les nœuds des périphériques du disque et évite les messages d'alerte du noyau au démarrage à propos de types de partitions inconnus.

2. Étiquettes de disque Sun (Sun disk labels)

Il faut porter une attention particulière aux systèmes SPARC car les disques contiennent des étiquettes de disque Sun.

2. Étiquettes de disque Sun (Sun disk labels)

L'organisation normale des disques munis d'étiquettes de disque Sun est de faire commencer la première partition à partir du bloc zéro du disque. Cela implique que l'étiquette du disque se trouve en fait sur sa première partition. Cela fonctionne parfaitement avec les systèmes de fichiers ext2 (et c'est essentiel pour amorcer un système avec avec SILO) mais ces partitions ne doivent pas être utilisées avec LVM car celui-ci commence à écrire au tout début du périphérique et écrase alors l'étiquette de disque.

Si vous voulez utiliser un disque avec des étiquettes de disque Sun, assurez-vous de faire commencer votre partition à partir du cylindre 1 ou supérieur.

Chapitre 13. Recettes

Ce chapitre détaille plusieurs « recettes » différentes pour mettre en place LVM. L'objectif est que le lecteur puisse adapter ces recettes pour son système et ses propres besoins.

1. Mettre en place LVM avec trois disques SCSI

Pour cette recette, on dispose de trois disques SCSI qui vont être mis dans un volume logique utilisant LVM. Les disques sont /dev/sda, /dev/sdb et /dev/sdc.

1.1. Préparation des disques

Il faut préparer le disque avant de pouvoir l'utiliser dans un groupe de volumes :

Attention !

Les commandes suivantes vont détruire toutes les données présentes sur /dev/sda, /dev/sdb et /dev/sdc.

Lancez pvcreate sur les disques :

```
# pvcreate /dev/sda
# pvcreate /dev/sdb
# pvcreate /dev/sdc
```

Cela crée une zone avec les descripteurs de groupes de volumes (volume group descriptor area, VG-DA) au début des disques.

1.2. Configuration du groupe de volumes

1. Créez un groupe de volumes

   ```
   # vgcreate mon_groupe_de_volumes /dev/sda /dev/sdb /dev/sdc
   ```

2. Utilisez vgdisplay pour vérifier le groupe de volume

   ```
   # vgdisplay
   --- Volume Group ---
   VG Name               mon_groupe_de_volumes
   VG Access             read/write
   VG Status             available/resizable
   VG #                  1
   MAX LV                256
   Cur LV                0
   Open LV               0
   MAX LV Size           255.99 GB
   Max PV                256
   Cur PV                3
   Act PV                3
   VG Size               1.45 GB
   PE Size               4 MB
   Total PE              372
   ```

```
Alloc PE / Size       0 / 0
Free  PE / Size       372/ 1.45 GB
VG UUID               nP2PY5-5TOS-hLx0-FDu0-2a6N-f37x-0BME0Y
```

Le plus important est de vérifier les trois premiers éléments et que la taille indiquée par « VG Size » est bien la taille de vos trois disques réunis.

1.3. Création du volume logique

Si le groupe de volumes semble correct, il est alors temps de créer un volume logique dessus.

Vous pouvez choisir la taille que vous voulez pour le volume logique (de la même manière que vous le feriez dans un environnement non LVM). Pour cet exemple, nous allons juste créer un volume logique d'1 Go dans le groupe de volumes. Nous n'utiliserons pas la répartition car il n'est pas actuellement possible de rajouter un disque une fois le volume logique créé.

```
# lvcreate -L1G -nmon_volume_logique mon_groupe_de_volumes
lvcreate -- doing automatic backup of "mon_groupe_de_volumes"
lvcreate -- logical volume "/dev/mon_groupe_de_volumes/mon_volume_logique" successfully created
```

1.4. Création du système de fichiers

Créez un système de fichiers ext2 dans le volume logique :

```
# mke2fs /dev/mon_groupe_de_volumes/mon_volume_logique
mke2fs 1.19, 13-Jul-2000 for EXT2 FS 0.5b, 95/08/09
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
131072 inodes, 262144 blocks
13107 blocks (5.00%) reserved for the super user
First data block=0
9 block groups
32768 blocks per group, 32768 fragments per group
16384 inodes per group
Superblock backups stored on blocks:
32768, 98304, 163840, 229376

Writing inode tables: done
Writing superblocks and filesystem accounting information: done
```

1.5. Test du système de fichiers

Montez le volume logique et vérifiez que tout semble correct.

```
# mount /dev/mon_groupe_de_volumes/mon_volume_logique /mnt
# df

Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/hda1              1311552    628824    616104  51% /
/dev/mon_groupe_de_volumes/mon_volume_logique
                       1040132        20    987276   0% /mnt
```

Si tout s'est passé correctement, vous avez désormais un volume logique avec un système de fichiers ext2 monté dans `/mnt`.

2. Mise en place de LVM sur trois disques SCSI avec répartition (striping)

Pour cette recette, on dispose de trois disques SCSI qui vont être mis dans un volume logique en utilisant LVM. Les disques sont /dev/sda, /dev/sdb et /dev/sdc.

Remarque

Il n'est pas actuellement possible d'ajouter un disque dans un volume logique réparti avec LVM 1. Utilisez LVM 2 avec le format de méta-données lvm 2 si vous voulez effectuer cela.

2.1. Préparation des partitions de disque

Il faut préparer le disque avant de pouvoir l'utiliser dans un groupe de volumes :

Attention !

Les commandes suivantes vont détruire toutes les données présentes sur /dev/sda, /dev/sdb et /dev/sdc.

Lancez pvcreate sur les disques :

```
# pvcreate /dev/sda
# pvcreate /dev/sdb
# pvcreate /dev/sdc
```

Cela crée une zone de descripteurs de groupes de volumes (volume group descriptor area, VGDA) au début des disques.

2.2. Configuration du groupe de volumes

1. Créez un groupe de volumes

   ```
   # vgcreate mon_groupe_de_volumes /dev/sda /dev/sdb /dev/sdc
   ```

2. Utilisez vgdisplay pour vérifier le groupe de volume

   ```
   # vgdisplay
   --- Volume Group ---
   VG Name                      mon_groupe_de_volumes
   VG Access             read/write
   VG Status             available/resizable
   VG #                  1
   MAX LV                256
   Cur LV                0
   Open LV               0
   MAX LV Size           255.99 GB
   Max PV                256
   Cur PV                3
   Act PV                3
   VG Size               1.45 GB
   PE Size               4 MB
   Total PE              372
   Alloc PE / Size       0 / 0
   ```

```
Free  PE / Size       372/ 1.45 GB
VG UUID               nP2PY5-5TOS-hLx0-FDu0-2a6N-f37x-0BME0Y
```

Le plus important est de vérifier les trois premiers éléments et que la taille indiquée par « VG Size » est bien la taille de vos trois disques réunis.

2.3. Création du volume logique

Si le groupe de volumes semble correct, il est alors temps de créer un volume logique dessus.

Vous pouvez choisir la taille que vous voulez pour le volume logique (de la même manière que vous le feriez sur un environnement non LVM). Pour cet exemple, nous allons juste créer un volume logique d'1 Go sur le groupe de volumes. Le volume logique sera réparti en utilisant une taille de bloc de 4 Ko. Cela devrait améliorer ses performances.

```
# lvcreate -i3 -I4 -L1G -nmon_volume_logique mon_groupe_de_volumes
lvcreate -- rounding 1048576 KB to stripe boundary size 1056768 KB / 258 PE
lvcreate -- doing automatic backup of "mon_groupe_de_volumes"
lvcreate -- logical volume "/dev/mon_groupe_de_volumes/mon_volume_logique" successfully created
```

Remarque

Si vous créez le volume logique avec l'option « -i2 », vous n'utiliserez que deux disques de votre groupe de volumes. Ceci est utile si vous voulez créer deux volumes logiques à partir du même volume physique, mais ce point n'est pas abordé dans cette recette.

2.4. Création du système de fichiers

Créez un système de fichiers ext2 dans le volume logique :

```
# mke2fs /dev/mon_groupe_de_volumes/mon_volume_logique
mke2fs 1.19, 13-Jul-2000 for EXT2 FS 0.5b, 95/08/09
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
132192 inodes, 264192 blocks
13209 blocks (5.00%) reserved for the super user
First data block=0
9 block groups
32768 blocks per group, 32768 fragments per group
14688 inodes per group
Superblock backups stored on blocks:
32768, 98304, 163840, 229376

Writing inode tables: done
Writing superblocks and filesystem accounting information: done
```

2.5. Test du système de fichiers

Montez le volume logique et vérifiez que tout semble correct.

```
# mount /dev/mon_groupe_de_volumes/mon_volume_logique /mnt
```

```
# df
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/hda1              1311552    628824    616104  51% /
/dev/mon_groupe_de_volumes/mon_volume_logique
                       1040132        20    987276   0% /mnt
```

Si tout s'est passé correctement, vous avez désormais un volume logique avec un système de fichiers ext2 monté dans `/mnt`.

3. Ajouter un nouveau disque dans un système avec plusieurs disques SCSI

3.1. Situation actuelle

Un serveur de données possède six disques organisés de la façon suivante :

```
# pvscan
pvscan -- ACTIVE   PV "/dev/sda"  of VG "dev"   [1.95 GB / 0 free]
pvscan -- ACTIVE   PV "/dev/sdb"  of VG "sales" [1.95 GB / 0 free]
pvscan -- ACTIVE   PV "/dev/sdc"  of VG "ops"   [1.95 GB / 44 MB free]
pvscan -- ACTIVE   PV "/dev/sdd"  of VG "dev"   [1.95 GB / 0 free]
pvscan -- ACTIVE   PV "/dev/sde1" of VG "ops"   [996 MB / 52 MB free]
pvscan -- ACTIVE   PV "/dev/sde2" of VG "sales" [996 MB / 944 MB free]
pvscan -- ACTIVE   PV "/dev/sdf1" of VG "ops"   [996 MB / 0 free]
pvscan -- ACTIVE   PV "/dev/sdf2" of VG "dev"   [996 MB / 72 MB free]
pvscan -- total: 8 [11.72 GB] / in use: 8 [11.72 GB] / in no VG: 0 [0]

# df
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/dev/cvs           1342492    516468    757828  41% /mnt/dev/cvs
/dev/dev/users         2064208   2060036      4172 100% /mnt/dev/users
/dev/dev/build         1548144   1023041    525103  66% /mnt/dev/build
/dev/ops/databases     2890692   2302417    588275  79% /mnt/ops/databases
/dev/sales/users       2064208    871214   1192994  42% /mnt/sales/users
/dev/ops/batch         1032088    897122    134966  86% /mnt/ops/batch
```

Comme vous pouvez le voir, les groupes « dev » et « ops » sont presque pleins. Un nouveau disque est acheté et ajouté au système. Il devient `/dev/sdg`.

3.2. Préparation du partitionnement du disque

Le nouveau disque doit être partagé entre ops et dev. Il est donc partitionné en deux volumes physiques /dev/sdg1 et /dev/sdg2 :

```
# fdisk /dev/sdg

Device contains neither a valid DOS partition table, nor Sun or SGI
disklabel Building a new DOS disklabel. Changes will remain in memory
only, until you decide to write them. After that, of course, the
previous content won't be recoverable.

Command (m for help): n
Command action
e   extended
p   primary partition (1-4)

p
Partition number (1-4): 1
```

```
First cylinder (1-1000, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-1000, default 1000): 500

Command (m for help): n
Command action
e   extended
p   primary partition (1-4)

p
Partition number (1-4): 2
First cylinder (501-1000, default 501):
Using default value 501
Last cylinder or +size or +sizeM or +sizeK (501-1000, default 1000):
Using default value 1000

Command (m for help): t
Partition number (1-4): 1
Hex code (type L to list codes): 8e
Changed system type of partition 1 to 8e (Unknown)

Command (m for help): t
Partition number (1-4): 2
Hex code (type L to list codes): 8e
Changed system type of partition 2 to 8e (Unknown)

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: If you have created or modified any DOS 6.x partitions,
please see the fdisk manual page for additional information.
```

Ensuite, les volumes physiques sont créés sur cette partition :

```
# pvcreate /dev/sdg1
pvcreate -- physical volume "/dev/sdg1" successfully created

# pvcreate /dev/sdg2
pvcreate -- physical volume "/dev/sdg2" successfully created
```

3.3. Ajout des nouvelles partitions au groupe de volumes

Les volumes sont ensuite ajoutés aux groupes de volumes dev et ops :

```
# vgextend ops /dev/sdg1
vgextend -- INFO: maximum logical volume size is 255.99 Gigabyte
vgextend -- doing automatic backup of volume group "ops"
vgextend -- volume group "ops" successfully extended

# vgextend dev /dev/sdg2
vgextend -- INFO: maximum logical volume size is 255.99 Gigabyte
vgextend -- doing automatic backup of volume group "dev"
vgextend -- volume group "dev" successfully extended

# pvscan
pvscan -- reading all physical volumes (this may take a while...)
pvscan -- ACTIVE   PV "/dev/sda"  of VG "dev"   [1.95 GB / 0 free]
pvscan -- ACTIVE   PV "/dev/sdb"  of VG "sales" [1.95 GB / 0 free]
pvscan -- ACTIVE   PV "/dev/sdc"  of VG "ops"   [1.95 GB / 44 MB free]
pvscan -- ACTIVE   PV "/dev/sdd"  of VG "dev"   [1.95 GB / 0 free]
```

```
pvscan -- ACTIVE   PV "/dev/sde1" of VG "ops"   [996 MB / 52 MB free]
pvscan -- ACTIVE   PV "/dev/sde2" of VG "sales" [996 MB / 944 MB free]
pvscan -- ACTIVE   PV "/dev/sdf1" of VG "ops"   [996 MB / 0 free]
pvscan -- ACTIVE   PV "/dev/sdf2" of VG "dev"   [996 MB / 72 MB free]
pvscan -- ACTIVE   PV "/dev/sdg1" of VG "ops"   [996 MB / 996 MB free]
pvscan -- ACTIVE   PV "/dev/sdg2" of VG "dev"   [996 MB / 996 MB free]
pvscan -- total: 10 [13.67 GB] / in use: 10 [13.67 GB] / in no VG: 0 [0]
```

3.4. Extension des systèmes de fichiers

L'action suivante est d'étendre les systèmes de fichiers pour que de l'espace disque supplémentaire soit disponible aux utilisateurs.

Il existe des outils qui permettent le redimensionnement à chaud des systèmes de fichiers ext2, mais nous utilisons ici la méthode la plus sûre et démontons les deux systèmes de fichiers avant le redimensionnement :

```
# umount /mnt/ops/batch
# umount /mnt/dev/users
```

Nous utilisons ensuite la commande e2fsadm pour redimensionner le volume logique et le système de fichiers ext2 en une seule opération. Nous utilisons ext2resize à la place de resize2fs (qui est la commande par défaut de e2fsadm). Pour cela, nous définissons la variable d'environnement E2FSADM_RESIZE_CMD pour dire à e2fsadm d'utiliser cette commande.

```
# export E2FSADM_RESIZE_CMD=ext2resize
# e2fsadm /dev/ops/batch -L+500M
e2fsck 1.18, 11-Nov-1999 for EXT2 FS 0.5b, 95/08/09
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/ops/batch: 11/131072 files (0.0<!--  non-contiguous), 4127/262144 blocks
lvextend -- extending logical volume "/dev/ops/batch" to 1.49 GB
lvextend -- doing automatic backup of volume group "ops"
lvextend -- logical volume "/dev/ops/batch" successfully extended

ext2resize v1.1.15 - 2000/08/08 for EXT2FS 0.5b
e2fsadm -- ext2fs in logical volume "/dev/ops/batch" successfully extended to 1.49 GB

# e2fsadm /dev/dev/users -L+900M
e2fsck 1.18, 11-Nov-1999 for EXT2 FS 0.5b, 95/08/09
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/dev/users: 12/262144 files (0.0% non-contiguous), 275245/524288 blocks
lvextend -- extending logical volume "/dev/dev/users" to 2.88 GB
lvextend -- doing automatic backup of volume group "dev"
lvextend -- logical volume "/dev/dev/users" successfully extended

ext2resize v1.1.15 - 2000/08/08 for EXT2FS 0.5b
e2fsadm -- ext2fs in logical volume "/dev/dev/users" successfully extended to 2.88 GB
```

3.5. Remontage des volumes étendus

Nous pouvons désormais remonter les systèmes de fichiers et vérifier qu'il y a maintenant plus d'espace libre.

```
# mount /dev/ops/batch
# mount /dev/dev/users
# df
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/dev/cvs           1342492    516468    757828  41% /mnt/dev/cvs
/dev/dev/users         2969360   2060036    909324  69% /mnt/dev/users
/dev/dev/build         1548144   1023041    525103  66% /mnt/dev/build
/dev/ops/databases     2890692   2302417    588275  79% /mnt/ops/databases
/dev/sales/users       2064208    871214   1192994  42% /mnt/sales/users
/dev/ops/batch         1535856    897122    638734  58% /mnt/ops/batch
```

4. Faire une sauvegarde avec un instantané (snapshot)

À la suite de l'exemple précédent, nous désirons maintenant utiliser l'espace supplémentaire dans le groupe de volumes « ops » pour faire une sauvegarde de la base de données tous les soirs. Pour nous assurer de la cohérence des données sur la bande, nous utilisons la fonctionnalité d'instantané de LVM.

Ce type de volume est une copie en lecture seule d'un autre volume et qui contient toutes les données qui étaient dans ce volume au moment de sa création. Cela signifie que nous pouvons faire une sauvegarde du volume sans se préoccuper des modifications de données pendant la sauvegarde et par conséquent il n'y a pas besoin de garder la base de données éteinte pendant la sauvegarde.

4.1. Créer le volume d'instantané

Il reste un peu plus de 500 Mo d'espace libre dans le groupe de volume « ops ». Nous allons l'utiliser entièrement pour le volume logique d'instantané. Un volume logique d'instantané peut être aussi grand ou petit que vous le souhaitez, mais il doit être assez grand pour contenir toutes les modifications qui peuvent apparaître sur le volume originel pendant la durée de vie de l'instantané. Ici, permettre 500 Mo de modifications dans la base de données devrait être très large.

```
# lvcreate -L592M -s -n dbbackup /dev/ops/databases
lvcreate -- WARNING: the snapshot must be disabled if it gets full
lvcreate -- INFO: using default snapshot chunk size of 64 KB for "/dev/ops/dbbackup"
lvcreate -- doing automatic backup of "ops"
lvcreate -- logical volume "/dev/ops/dbbackup" successfully created
```

Note

Si l'instantané est effectué pour un système de fichiers XFS, la commande xfs_freeze doit être utilisée pour figer le système de fichiers avant la création de l'instantané. (Si le système de fichiers est monté.)

```
# xfs_freeze -f /mnt/point; lvcreate -L592M -s -n dbbackup /dev/ops/databases; xfs_freeze -u /mnt/point
```

Les instantanés pleins sont automatiquement désactivés

Si le volume logique d'instantané devient plein, il deviendra inutilisable. Il est donc vital d'allouer suffisamment d'espace.

4.2. Monter le volume d'instantané

Nous pouvons désormais créer un point de montage et monter le volume.

```
# mkdir /mnt/ops/dbbackup
# mount /dev/ops/dbbackup /mnt/ops/dbbackup

mount: block device /dev/ops/dbbackup is write-protected, mounting read-only
```

Si vous utilisez le système de fichiers XFS, il faut ajouter l'option `nouuid` à la commande mount :

```
# mount /dev/ops/dbbackup /mnt/ops/dbbackup -onouuid,ro
```

Note

Avant, on conseillait l'option norecovery pour pouvoir monter des instantanés XFS. Désormais, il est recommandé de ne pas utiliser cette option mais, à la place, d'utiliser xfs_freeze pour figer le système de fichiers avant la création de l'instantané.

4.3. Faire la sauvegarde

Je suppose que vous avez des stratégies de sauvegarde un peu plus sophistiquées que ça !

```
# tar -cf /dev/rmt0 /mnt/ops/dbbackup
tar: Removing leading `/' from member names
```

4.4. Supprimer l'instantané

Une fois la sauvegarde effectuée, vous pouvez démonter le volume et le supprimer du système. Vous devriez enlever les volumes d'instantanés dès que vous avez fini de les utiliser car il font une copie de toutes les données écrites sur le volume originel et cela peut impacter les performances.

```
# umount /mnt/ops/dbbackup
# lvremove /dev/ops/dbbackup

lvremove -- do you really want to remove "/dev/ops/dbbackup"? [y/n]: y
lvremove -- doing automatic backup of volume group "ops"
lvremove -- logical volume "/dev/ops/dbbackup" successfully removed
```

5. Enlever un disque

Admettons que vous ayez un vieux disque IDE sur /dev/hdb. Vous voulez le retirer mais de nombreux fichiers sont encore dessus.

Sauvegarder votre système

Vous devriez toujours effectuer une sauvegarde de votre système avant de faire une opération avec pvremove.

5.1. Distribuer les anciens extents vers des disques existants du groupe de volumes

Si vous avez assez d'extents libres sur les autres disques du groupe de volumes, c'est très simple. Il suffit d'exécuter la commande :

```
# pvmove /dev/hdb
pvmove -- moving physical extents in active volume group "dev"
pvmove -- WARNING: moving of active logical volumes may cause data loss!
pvmove -- do you want to continue? [y/n] y

pvmove -- 249 extents of physical volume "/dev/hdb" successfully moved
```

Cela va déplacer les extents physiques alloués sur /dev/hdb vers les autres disques du groupe de volumes.

pvmove est lent

Attention, pvmove est très lent car il doit copier le contenu d'un disque vers un ou plusieurs autres, bloc par bloc. Si vous souhaitez plus d'informations sur l'avancement de l'opération, utilisez l'option `-v` avec pvmove.

5.1.1. Enlever l'ancien disque

Nous pouvons désormais supprimer l'ancien disque du groupe de volumes.

```
# vgreduce dev /dev/hdb
vgreduce -- doing automatic backup of volume group "dev"
vgreduce -- volume group "dev" successfully reduced by physical volume:
vgreduce -- /dev/hdb
```

Le disque peut désormais être enlevé physiquement au prochain arrêt de la machine ou bien être réalloué à d'autres utilisateurs.

5.2. Distribuer les anciens extents sur un nouveau disque de remplacement

Si vous n'avez pas assez d'extents physiques libres pour les anciens extents, il va falloir rajouter un disque dans le groupe de volumes et y déplacer les extents.

5.2.1. Préparer le disque

D'abord, il faut utiliser pvcreate sur le nouveau disque pour le rendre disponible dans LVM. Dans cette recette, nous montrons qu'il n'est pas nécessaire de partitionner un disque pour pouvoir l'utiliser.

```
# pvcreate /dev/sdf
pvcreate -- physical volume "/dev/sdf" successfully created
```

5.2.2. Ajouter le disque au groupe de volumes

Comme les développeurs utilisent beaucoup d'espace disque, leur volume est un bon candidat pour l'ajout.

```
# vgextend dev /dev/sdf
vgextend -- INFO: maximum logical volume size is 255.99 Gigabyte
vgextend -- doing automatic backup of volume group "dev"
```

```
vgextend -- volume group "dev" successfully extended
```

5.2.3. Déplacer les données

Ensuite, nous déplaçons les données de l'ancien disque vers le nouveau. Remarquez qu'il n'est pas nécessaire de démonter le système de fichiers avant de le faire. Cependant, il est *vivement* recommandé de faire une sauvegarde complète avant de faire cette opération au cas où elle soit interrompue par une coupure électrique ou tout autre problème. La commande pvmove peut prendre un temps considérable et réduire les performances des deux volumes. Il est donc conseillé d'attendre qu'ils ne soient pas trop occupés.

```
# pvmove /dev/hdb /dev/sdf
pvmove -- moving physical extents in active volume group "dev"
pvmove -- WARNING: moving of active logical volumes may cause data loss!
pvmove -- do you want to continue? [y/n] y
pvmove -- 249 extents of physical volume "/dev/hdb" successfully moved
```

5.2.4. Enlever l'ancien disque

Nous pouvons désormais supprimer l'ancien disque du groupe de volumes.

```
# vgreduce dev /dev/hdb
vgreduce -- doing automatic backup of volume group "dev"
vgreduce -- volume group "dev" successfully reduced by physical volume:
vgreduce -- /dev/hdb
```

Le disque peut être enlevé physiquement au prochain arrêt de la machine ou bien être réalloué à d'autres utilisateurs.

6. Déplacer un groupe de volumes vers un autre système

Il est très simple de déplacer un groupe de volumes entier vers un autre système. Par exemple, si on acquiert un nouveau serveur pour le département. Pour cela, on utilise les commandes vgexport et vgimport.

Note

vgexport et vgimport ne sont pas nécessaires pour déplacer des disques. Ce sont des outils d'administration permettant d'empêcher les accès aux volumes le temps de les déplacer.

6.1. Démonter le système de fichiers

Il faut nous assurer qu'aucun utilisateur n'accède aux fichiers sur le volume et ensuite il faut le démonter :

```
# unmount /mnt/design/users
```

6.2. Marquer le groupe de volumes comme inactif

Rendre le groupe de volumes inactif l'enlève du noyau et empêche toute activité sur celui-ci.

```
# vgchange -an design
vgchange -- volume group "design" successfully deactivated
```

6.3. Exporter le groupe de volumes

Il est ensuite nécessaire d'exporter le groupe de volumes. Cela empêche qu'on y accède sur « l'ancienne » machine et le prépare à être retiré.

```
# vgexport design
vgexport -- volume group "design" sucessfully exported
```

Au prochain arrêt de la machine, le disque peut être enlevé et placé sur sa nouvelle machine.

6.4. Importer le groupe de volumes

Une fois connecté au nouveau système, un pvscan initial montre qu'il est devenu /dev/sdb :

```
# pvscan
pvscan -- reading all physical volumes (this may take a while...)
pvscan -- inactive PV "/dev/sdb1"  is in EXPORTED VG "design" [996 MB / 996 MB free]
pvscan -- inactive PV "/dev/sdb2"  is in EXPORTED VG "design" [996 MB / 244 MB free]
pvscan -- total: 2 [1.95 GB] / in use: 2 [1.95 GB] / in no VG: 0 [0]
```

Nous pouvons désormais importer le groupe de volumes (ce qui l'active au passage) et monter le système de fichiers.

```
# vgimport design /dev/sdb1 /dev/sdb2
vgimport -- doing automatic backup of volume group "design"
vgimport -- volume group "design" successfully imported and activated
```

6.5. Activer le groupe de volumes

Vous devez activer le groupe de volumes avant de pouvoir y accéder.

```
# vgchange -ay design
```

6.6. Monter le système de fichiers

```
# mkdir -p /mnt/design/users
# mount /dev/design/users /mnt/design/users
```

Le système de fichiers est désormais prêt à être utilisé.

7. Diviser un groupe de volumes

Supposons que nous devons rajouter le nouveau groupe d'utilisateurs « design » au système. Un moyen de le gérer est d'ajouter un groupe de volumes au système pour y mettre leurs données. Ici, nous n'avons pas de nouveau disque mais beaucoup d'espace libre sur les disques existants qui peut être réalloué.

7.1. Déterminer l'espace libre

```
# pvscan
pvscan -- reading all physical volumes (this may take a while...)
pvscan -- ACTIVE   PV "/dev/sda"  of VG "dev"   [1.95 GB / 0 free]
pvscan -- ACTIVE   PV "/dev/sdb"  of VG "sales" [1.95 GB / 1.27 GB free]
pvscan -- ACTIVE   PV "/dev/sdc"  of VG "ops"   [1.95 GB / 564 MB free]
pvscan -- ACTIVE   PV "/dev/sdd"  of VG "dev"   [1.95 GB / 0 free]
pvscan -- ACTIVE   PV "/dev/sde"  of VG "ops"   [1.95 GB / 1.9 GB free]
pvscan -- ACTIVE   PV "/dev/sdf"  of VG "dev"   [1.95 GB / 1.33 GB free]
pvscan -- ACTIVE   PV "/dev/sdg1" of VG "ops"   [996 MB / 432 MB free]
pvscan -- ACTIVE   PV "/dev/sdg2" of VG "dev"   [996 MB / 632 MB free]
pvscan -- total: 8 [13.67 GB] / in use: 8 [13.67 GB] / in no VG: 0 [0]
```

Nous décidons de réallouer /dev/sdg1 et /dev/sdg2 à « design ». Il faut commencer par déplacer tous les extents physiques dans des espaces libres d'autres volumes (dans notre cas /dev/sdf pour le VG dev et /dev/sde pour ops).

7.2. Déplacer les données des disques à réutiliser

De l'espace est toujours utilisé sur les volumes choisis, il faut donc déplacer les données autre part.

Déplacez tous les extents physiques utilisés de /dev/sdg1 vers /dev/sde et de /dev/sdg2 vers /dev/sdf.

```
# pvmove /dev/sdg1 /dev/sde
pvmove -- moving physical extents in active volume group "ops"
pvmove -- WARNING: moving of active logical volumes may cause data loss!
pvmove -- do you want to continue? [y/n] y
pvmove -- doing automatic backup of volume group "ops"
pvmove -- 141 extents of physical volume "/dev/sdg1" successfully moved

# pvmove /dev/sdg2 /dev/sdf
pvmove -- moving physical extents in active volume group "dev"
pvmove -- WARNING: moving of active logical volumes may cause data loss!
pvmove -- do you want to continue? [y/n] y
pvmove -- doing automatic backup of volume group "dev"
pvmove -- 91 extents of physical volume "/dev/sdg2" successfully moved
```

7.3. Créer un nouveau groupe de volumes

Maintenant, séparez /dev/sdg2 de dev et ajoutez-le à un nouveau groupe appelé « design ». Il est possible de faire cela avec vgreduce et vgcreate, mais vgsplit combine les deux commandes.

```
# vgsplit dev design /dev/sdg2
vgsplit -- doing automatic backup of volume group "dev"
vgsplit -- doing automatic backup of volume group "design"
vgsplit -- volume group "dev" successfully split into "dev" and "design"
```

7.4. Enlever le volume restant

Ensuite, enlevez /dev/sdg1 de ops et ajoutez-le dans design.

```
# vgreduce ops /dev/sdg1
vgreduce -- doing automatic backup of volume group "ops"
vgreduce -- volume group "ops" successfully reduced by physical volume:
```

```
vgreduce -- /dev/sdg1

# vgextend design /dev/sdg1
vgextend -- INFO: maximum logical volume size is 255.99 Gigabyte
vgextend -- doing automatic backup of volume group "design"
vgextend -- volume group "design" successfully extended
```

7.5. Créer le nouveau volume logique

Maintenant, créez un volume logique. Au lieu d'allouer tout l'espace disponible, gardez un peu d'espace de réserve au cas où vous en auriez besoin ailleurs.

```
# lvcreate -L750M -n users design
lvcreate -- rounding up size to physical extent boundary "752 MB"
lvcreate -- doing automatic backup of "design"
lvcreate -- logical volume "/dev/design/users" successfully created
```

7.6. Créez un système de fichiers sur le volume

```
# mke2fs /dev/design/users
mke2fs 1.18, 11-Nov-1999 for EXT2 FS 0.5b, 95/08/09
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
96384 inodes, 192512 blocks
9625 blocks (5.00<!-- ) reserved for the super user
First data block=0
6 block groups
32768 blocks per group, 32768 fragments per group
16064 inodes per group
Superblock backups stored on blocks:
32768, 98304, 163840

Writing inode tables: done
Writing superblocks and filesystem accounting information: done
```

7.7. Montez le nouveau volume

```
# mkdir -p /mnt/design/users
# mount /dev/design/users /mnt/design/users/
```

C'est aussi une bonne d'idée d'ajouter une entrée dans /etc/fstab pour ce système de fichiers :

```
/dev/design/user /mnt/design/users   ext2    defaults        1 2
```

8. Convertir le système de fichiers racine en LVM 1

Sauvegardez votre système

Il est fortement recommandé de faire une sauvegarde complète de votre système avant d'essayer de convertir votre racine en LVM 1.

Complications des mises à jour

Avoir son système de fichiers racine en LVM 1 peut compliquer significativement les procédures de mise à jour (suivant les distributions) donc la décision ne doit pas être prise à la légère. Plus précisément, vous devez vous assurer que le module LVM 1 (s'il n'est pas compilé directement dans le noyau) ainsi que les utilitaires vgscan et vgchange sont disponibles avant, pendant et après la mise à jour.

Complications des récupérations

Avoir son système racine sous LVM 1 peut compliquer significativement la récupération de systèmes de fichiers endommagés. Si vous perdez votre initrd, il sera très difficile de démarrer le système. Il faudra alors un disque de récupération contenant le noyau, le module et les utilitaires LVM 1 et tous les outils nécessaires pour récupérer un système de fichiers endommagé. Soyez sûr de faire des sauvegardes régulières et de garder une méthode de démarrage alternative qui permet la récupération de LVM.

Dans cet exemple, tout le système était installé sur une seule partition racine excepté /boot. Le système avait 2 Go de disque partitionnés de la façon suivante :

```
/dev/hda1  /boot
/dev/hda2  swap
/dev/hda3  /
```

La partition / remplissait tout le disque hormis la partition /boot et le swap. Un des pré-requis important pour cette procédure est que moins de la moitié de la partition racine soit remplie (pour pouvoir en faire une copie dans le volume logique). Si ce n'est pas le cas, un second disque devra être utilisé. La procédure est alors similaire, sauf qu'il n'est pas nécessaire de réduire la partition racine existante et /dev/hda4 doit être remplacé, par exemple, par /dev/hdb1 dans notre cas.

Le plus simple pour faire cela est d'utiliser GNU parted. Ce logiciel permet de réduire ou d'agrandir des partitions avec des systèmes de fichiers. Il est possible d'utiliser resize2fs et fdisk, mais GNU parted est moins sensible aux erreurs. S'il n'est pas inclus dans votre distribution, vous pouvez le télécharger à l'adresse : ftp://ftp.gnu.org/pub/gnu/parted.

Une fois que vous avez installé parted et QUE VOUS AVEZ SAUVEGARDÉ VOTRE SYSTÈME :

8.1. Démarrer en mode mono-utilisateur

Démarrez en mode mono-utilisateur (tapez **linux S** à l'invite de Lilo). Ceci est important car cela assure que le système de fichiers racine sera monté en lecture seule et qu'aucun programme n'accédera au disque.

8.2. Utiliser Parted

Utilisez parted pour réduire la partition racine. Cela doit permettre d'avoir assez de place sur le disque pour une copie complète de celui-ci dans un volume logique. Dans cet exemple, la partition de 1,8 Go est réduite à 1 Go. Cela montre les tailles et noms des partitions sur le disque.

```
# parted /dev/hda
(parted) p
.
.
.
```

Maintenant, retaillez la partition :

```
(parted) resize 3 145 999
```

Le premier nombre est le numéro de la partition (hda3). Le second est le numéro de début de partition de hda3 (en Mo depuis le début du disque), ne le changez pas. Le dernier numéro (en Mo depuis le début du disque) indique la fin de la partition et devrait la réduire d'à peu près à la moitié de sa taille actuelle.

Créez une nouvelle partition :

```
(parted) mkpart primary ext2 1000 1999
```

Cela crée une nouvelle partition pour contenir les données initiales de LVM 1. Elle devrait commencer juste à la suite de la partition réduite hda3 et terminer à la fin du disque.

Quittez parted :

```
(parted) q
```

8.3. Redémarrer

Redémarrez le système.

8.4. Vérifier les options de configuration du noyau

Assurez-vous que le noyau qui tourne actuellement fonctionne avec LVM 1 et est configuré avec les paramètres CONFIG_BLK_DEV_RAM et CONFIG_BLK_DEV_INITRD.

8.5. Ajuster le type de partition

Changez le type de la nouvelle partition de Linux à LVM (8e). Parted ne connaît pas les partitions LVM 1, donc il faut le faire avec fdisk.

```
# fdisk /dev/hda
Command (m for help): t
Partition number (1-4): 4
Hex code (type L to list codes): 8e
Changed system type of partition 4 to 8e (Unknown)
Command (m for help): w
```

8.6. Configurer LVM 1 pour la nouvelle configuration

- Initialisez LVM 1 (vgscan) :

  ```
  # vgscan
  ```

- Insérez la nouvelle partition dans un PV :

  ```
  # pvcreate /dev/hda4
  ```

- Créez un nouveau groupe de volumes :

```
# vgcreate vg /dev/hda4
```

- Créez un volume logique pour contenir la nouvelle racine :

```
# lvcreate -L250M -n root vg
```

8.7. Créer le système de fichiers

Créez un système de fichiers dans le volume logique et copiez-y les fichiers racines.

```
# mke2fs /dev/vg/root
# mount /dev/vg/root /mnt/
# find / -xdev | cpio -pvmd /mnt
```

8.8. Mise à jour de /etc/fstab

Ouvrez /mnt/etc/fstab de la nouvelle racine pour monter / sur /dev/vg/root. Par exemple :

```
/dev/hda3       /    ext2       defaults 1 1
```

devient :

```
/dev/vg/root    /    ext2       defaults 1 1
```

8.9. Créez une image disque RAM d'initialisation pour LVM 1

```
# lvmcreate_initrd
```

Retenez le nom du fichier créé par lvmcreate_initrd. Il doit être dans /boot.

8.10. Mise à jour de /etc/lilo.conf

Ajoutez une entrée dans /etc/lilo.conf pour LVM 1. Cela devrait ressembler à ceci :

```
image   = /boot/KERNEL_IMAGE_NAME
label   = lvm
root    = /dev/vg/root
initrd  = /boot/INITRD_IMAGE_NAME
ramdisk = 8192
```

où KERNEL_IMAGE_NAME est le nom de votre noyau avec LVM 1 et INITRD_IMAGE_NAME est le nom de l'image initrd créée par lvmcreate_initrd. Il se peut que la valeur pour ramdisk doive être augmentée si vous avez une grosse configuration LVM 1, mais 8192 devrait suffire pour la plupart des utilisateurs. La valeur par défaut de ramdisk est 4096. Dans le doute, vérifiez dans la sortie de la commande lvmcreate_initrd la ligne qui dit :

```
lvmcreate_initrd -- making loopback file (6189 kB)
```

et créez un ramdisk de la taille entre parenthèses.

Vous devez recopier ce nouveau lilo.conf dans le nouveau système de fichiers racine.

```
# cp /etc/lilo.conf /mnt/etc/
```

8.11. Lancez lilo pour écrire le nouveau secteur de boot

```
# lilo
```

8.12. Redémarrer avec LVM

Redémarrez et tapez « lvm » à l'invite de Lilo. Le système devrait démarrer sous Linux en utilisant le nouveau volume logique.

Si cela a fonctionné, vous devriez mettre lvm comme démarrage par défaut en ajoutant la ligne suivante dans la première section de /etc/lilo.conf :

```
default=lvm
```

Si cela n'a pas fonctionné, redémarrez normalement et penchez-vous sur le problème. Cela peut provenir d'une erreur de syntaxe dans lilo.conf ou de LVM 1 qui n'est pas disponible dans l'image disque RAM ou le noyau. Examinez soigneusement le message affiché au démarrage.

8.13. Ajoutez le reste du disque dans LVM

Une fois que vous êtes satisfait de la configuration, vous pouvez ajoutez le reste du disque dans LVM 1, y compris l'ancienne partition racine.

D'abord, il faut mettre le type de partition à 8e (LVM) :

```
# fdisk /dev/hda
Command (m for help): t
Partition number (1-4): 3
Hex code (type L to list codes): 8e
Changed system type of partition 3 to 8e (Unknown)
Command (m for help): w
```

Convertissez-la en PV et ajoutez-la au groupe de volumes :

```
# pvcreate /dev/hda3
# vgextend vg /dev/hda3
```

Annexe A. Opérations dangereuses

Attention

Ne faites pas ceci si vous n'êtes pas sûr de ce que vous faites. Vous y perdrez probablement toutes vos données.

1. Restaurer les UUID de VG avec uuid_fixer

Si vous avez mis à jour LVM d'une version précédente aux premières versions 0.9 ou 0.9.1 et que **vgscan** indique vgscan -- no volume groups found, voici un moyen de corriger le problème :

- Téléchargez le programme de réparation des UUID à partir du répertoire des contributions chez Sistina.

  Il se trouve à : ftp://ftp.sistina.com/pub/LVM/contrib/uuid_fixer-0.3-IOP10.tar.gz.

- Extrayez `uuid_fixer-0.3-IOP10.tar.gz` :

  ```
  # tar zxf uuid_fixer-0.3-IOP10.tar.gz
  ```

- Allez dans uuid_fixer :

  ```
  # cd uuid_fixer
  ```

  Il y a deux possibilités à ce niveau :

  1. Utiliser le binaire déjà compilé (pour une architecture i386 uniquement)

     Assurez-vous de lister tous les PV dans le VG que vous restaurez et suivez les indications :

     ```
     # ./uuid_fixer <LISTE DE TOUS LES PV DU VG QUI DOIVENT ÊTRE RESTAURÉS>
     ```

  2. Construire le programme uuid_fixer à partir des sources

     Ouvrez le Makefile avec votre éditeur favori et vérifiez que LVMDIR pointe sur vos sources de LVM.

     Ensuite, lancez make.

     ```
     # make
     ```

     Lancez maintenant uuid_fixer. Assurez-vous de lister tous les PV dans le VG que vous restaurez et suivez les indications :

     ```
     # ./uuid_fixer <LISTE DE TOUS LES PV DU VG QUI DOIVENT ÊTRE RESTAURÉS>
     ```

- Désactivez les groupes de volumes (*optionnel*) :

  ```
  # vgchange -an
  ```

- Lancez vgscan :

```
# vgscan
```

- Réactivez des groupes de volumes :

```
# vgchange -ay
```

2. Partager des volumes LVM

LVM n'est pas conscient des clusters

Faites très attention en faisant cela car LVM n'est pas conscient des clusters et il est très facile de perdre toutes ses données.

Si vous avez un environnement avec du fiber-channel ou du SCSI partagé où plusieurs machines peuvent avoir un accès physique à un lot de disques, vous pouvez utiliser LVM pour diviser ces disques en volumes logiques. Si vous voulez partager des données, vous devriez vraiment regarder GFS [http://www.redhat.com/software/rha/gfs] ou un autre système de fichiers pour cluster.

Le point clé à se souvenir lorsqu'on partage des volumes est que toute l'administration de LVM doit être effectuée sur l'un des nœuds et que LVM doit être arrêté sur tous les autres nœuds avant de faire une quelconque modification sur le nœud d'administration. Une fois les changements effectués, il est nécessaire de lancer vgscan sur les autres nœuds avant de réutiliser les groupes de volumes. Aussi, à moins que vous n'utilisiez un système de fichiers ou une application prenant en compte les clusters (comme GFS), un seul nœud à la fois peut monter un système de fichiers. C'est à l'administrateur de gérer cela ; LVM ne vous empêchera pas de corrompre vos données.

La séquence de démarrage de chaque nœud est la même que pour une configuration à un seul nœud avec l'utilisation de :

```
vgscan
vgchange -ay
```

dans les scripts de démarrage.

Si vous avez besoin d'apporter une **quelconque** modification aux méta-données de LVM (indépendamment que cela affecte ou pas les volumes montés sur d'autres nœuds), il faut utiliser la procédure suivante. Dans les points suivants, le « nœud d'administration » est un nœud choisi arbitrairement dans le cluster.

```
Nœud d'administration        Autres nœuds
----------------------       ------------------
                             Fermer tous les volumes logiques (umount)
                             vgchange -an
<faire les modifications,
par ex. lvextend>
                             vgscan
                             vgchange -ay
```

Les VG doivent être actifs sur le nœud d'administration

Vous n'avez pas besoin et ne devez pas défaire les VG du nœud d'administration. Par conséquent, cela doit être le nœud avec les pré-requis d'uptime les plus grands.

Je me répète encore : **Soyez très prudent en faisant cela.**

Annexe B. Signaler des erreurs ou des bogues

ATTENTION, toute communication avec les listes de diffusion de LVM doit se faire exclusivement en anglais.

Juste nous dire que LVM ne fonctionne pas ne nous donne pas assez d'informations pour vous aider. Nous devons connaître les différents composants de votre configuration. La première chose à faire est de regarder les archives de la liste de diffusion linux-lvm [http://www.redhat.com/archives/linux-lvm/] pour vérifier si quelqu'un n'a pas déjà signalé le même problème. Si vous ne trouvez pas de rapport de bogue pour un problème similaire au vôtre, vous devriez essayer de récupérer autant d'information que possible. La liste est organisée avec trois catégories d'erreurs :

- Pour les erreurs de compilation :
    1. Détaillez la version spécifique de LVM que vous utilisez. Si elle provient d'un paquet, donnez le nom du fichier tar et des différents correctifs que vous auriez pu appliquer. Si elle provient du serveur CVS public, donner la date et l'heure où vous l'avez récupérée.
    2. Donnez le message d'erreur exact. Fournissez aussi les quelques lignes qui précèdent le message d'erreur et les quelques lignes suivantes. Elles donnent parfois des indices sur la raison du problème.
    3. Listez les étapes, dans l'ordre, qui ont amené à l'erreur. Indiquez si l'erreur est reproductible. Si vous repartez de l'état initial, les mêmes étapes permettent-elles de reproduire l'erreur ?
- Pour les erreurs LVM :
    1. Incluez toutes les informations demandées dans le paragraphe compilation.
    2. Ajoutez une brève description de votre matériel : type de machine et de disques, interface des disques (SCSI, FC, NBD) et toute autre information matérielle que vous trouvez importante.
    3. Donnez les lignes de commandes de LVM qui ont amené l'erreur.
    4. Un fichier de trace produit par les commandes fautives. Pour cela, vérifiez la présence des lignes suivantes dans `/etc/lvm/lvm.conf` :

    ```
    log {
            file="/tmp/lvm2.log"
            level=7
            activation=1
    }
    ```
- Quand LVM panique :
    1. Incluez toutes les informations demandées dans les deux paragraphes précédents.
    2. Fournissez les informations de debug de la machine. Le mieux est de récupérer ces informations d'une console sur le port série de la machine, étant donné qu'il est difficile de faire un copier/coller d'une machine en panique et qu'il est très facile de faire une erreur en recopiant les informations à la main.

Cela peut faire beaucoup d'informations. Si vous avez plusieurs fichiers, faites une archive tar et gzip en un seul fichier. Envoyez un lien vers l'archive sur la liste de diffusion appropriée (voir la Section 1, « Listes de diffusion ») avec une brève description de l'erreur. Si vous n'avez pas de site web ou ftp pour mettre votre archive à disposition, vous pouvez essayer d'envoyer le fichier à la liste.

Annexe C. Contacts et liens

1. Listes de diffusion

ATTENTION, toute communication avec les listes de diffusion de LVM doit se faire exclusivement en anglais.

Avant de poster sur une de nos listes, veuillez lire tout le document et vérifier les archives pour voir si votre question n'a pas déjà eu une réponse. S'il vous plaît, ne postez que du format texte sur nos listes. Les messages enjolivés sont presque impossibles à lire s'ils ne sont pas lus sur un client de messagerie qui les comprend. L'étiquette standard des listes de diffusion s'applique. Les questions ou données de configuration incomplètes rendent les réponses très difficiles à faire.

Listes de diffusion de LVM

linux-lvm
: Cette liste est destinée aux questions et commentaires des utilisateurs. Vous pouvez obtenir vos réponses d'autres utilisateurs qui ont eu les mêmes problèmes. Les discussions ouvertes sont encouragées. Les rapports de bogues doivent être envoyés à cette liste.

  L'inscription se fait par une interface web [http://www.redhat.com/mailman/listinfo/linux-lvm].

  Les anciens messages sont consultables dans les archives [http://www.redhat.com/archives/linux-lvm/].

dm-devel
: Cette liste n'est pas spécifique à LVM mais elle est mentionnée ici du fait de l'utilisation du mappeur de périphériques par LVM 2.

  L'inscription se fait par l'interface web [http://www.redhat.com/mailman/listinfo/dm-devel].

  Les anciens messages sont consultables dans les archives [http://www.redhat.com/archives/dm-devel/].

Listes LVM relatives au commit

lvm2-commit
: Cette liste reçoit des messages automatiquement quand quelqu'un fait un commit dans le CVS de LVM 2. Son but est de permettre de suivre l'évolution de l'arborescence CVS.

  Les anciens messages sont consultables dans les archives [http://sources.redhat.com/ml/lvm2-cvs/].

lvm-commit
: Cette liste reçoit des messages automatiquement quand quelqu'un fait un commit dans le CVS de LVM. Son but est de permettre de suivre l'évolution de l'arborescence CVS.

  Les anciens messages sont consultables dans les archives [http://sources.redhat.com/ml/lvm-cvs/].

dm-commit
: Cette liste reçoit des messages automatiquement quand quelqu'un fait un commit dans le CVS de dm. Son but est de permettre de suivre l'évolution de l'arborescence CVS.

  Les anciens messages sont consultables dans les archives [http://sources.redhat.com/ml/dm-cvs/].

Anciennes listes

lvm-devel
: Cette liste n'est plus en activité, veuillez utiliser linux-lvm pour les discussions sur le développement de LVM.

lvm-bugs
: Cette liste n'est plus en activité. Les rapports d'anomalie doivent être envoyés à la liste linux-lvm.

2. Liens

Liens sur LVM :

- Page principale de Logical Volume Manager [http://sources.redhat.com/lvm2/].
- Page principale de LVM 1 [http://sources.redhat.com/lvm/].
- Page principale de Device-Mapper [http://sources.redhat.com/dm/].
- Site FTP de LVM 2 [ftp://sources.redhat.com/pub/lvm2/].
- Site FTP de LVM 1 [ftp://sources.redhat.com/pub/lvm/].
- Site FTP de Device-Mapper [ftp://sources.redhat.com/pub/dm/].

Annexe D. GNU Free Documentation License

Version 1.2, November 2002

1. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

2. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

3. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

4. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

5. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

6. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

7. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

8. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

9. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

10. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

11. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

12. ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.