

Utilizando LVM no Debian

Projeto Libertas-BR

<http://www.libertasbr.org.br>

8 de setembro de 2005



1 O que é o LVM

LVM (*Logical Volume Management*) é um “Gerenciador de Volumes Lógicos” implementado para o Linux. Sua implementação segue os mesmos padrões da implementação para HP-UX.

Esse *Gerenciamento de Volumes Lógicos* provê uma visão em nível mais alto do armazenamento de dados em disco do que a visão tradicional de partições e pontos de montagem; o que garante ao administrador muito mais flexibilidade na alocação de espaço no seu sistema. Os volumes de armazenamento criados podem ser redimensionados de acordo com as necessidades desse administrador.

Outra vantagem bastante interessante do uso do LVM, é a possibilidade de se mover os dados de um disco para outro com os usuários utilizando os dados. Além de diminuir o tempo com o sistema desligado, o LVM facilita toda a tarefa de administração do disco.

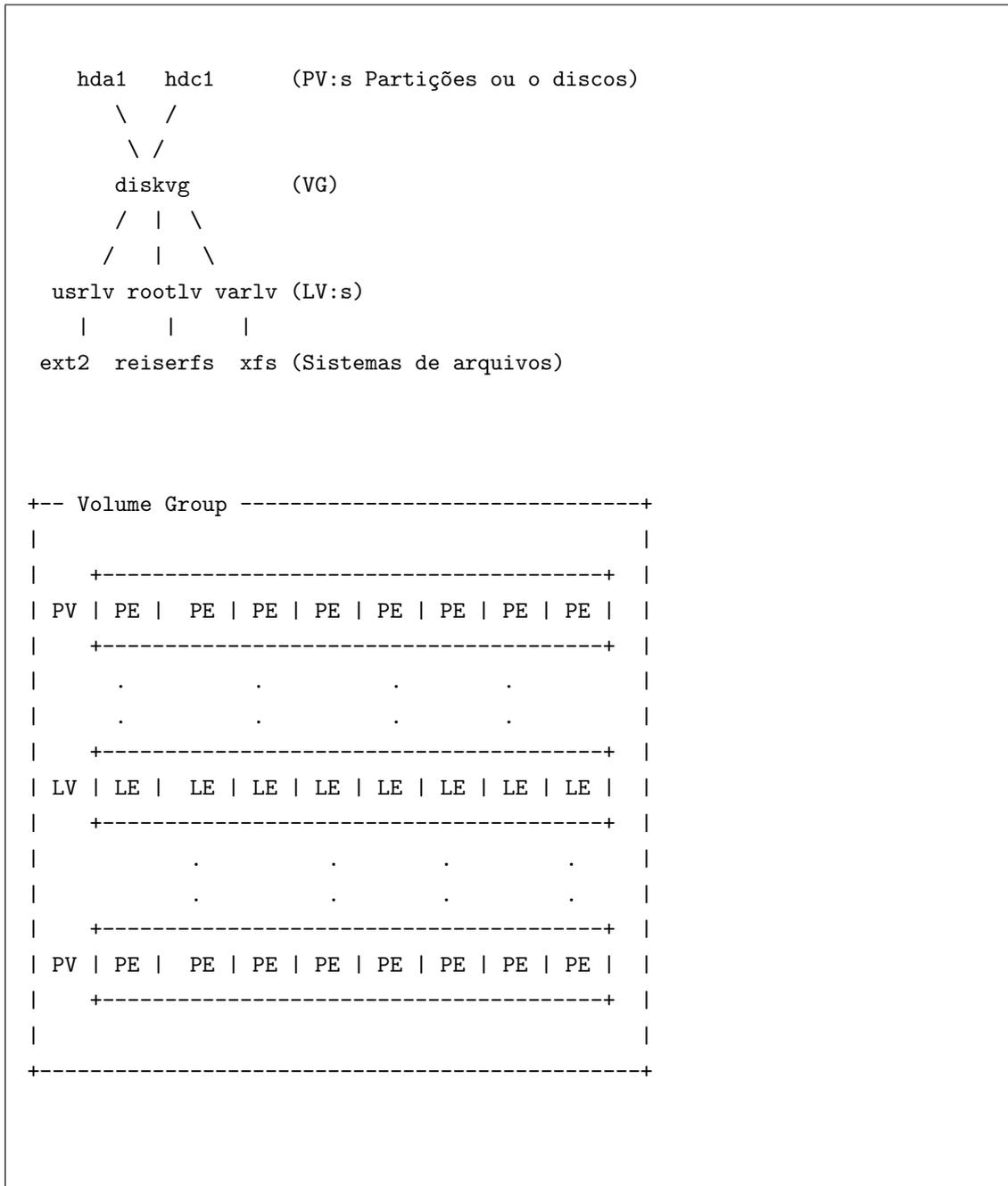
2 Conceitos Básicos

Para entender melhor o LVM, alguns conceitos devem ser explicados. Os elementos principais do LVM são:

- *physical volume* (PV): Pode ser um disco ou uma partição. Em geral, tudo que é visto pelo sistema como um disco pode ser um PV.
- *logical volume* (LV): Seria o equivalente a uma partição em um sistema sem LVM. É visto como um *block device* (dispositivo de blocos) pelo sistema e pode conter um sistema de arquivo (por exemplo /home).
- *volume group* (VG): Um VG é um grupo de volumes. Ele é responsável por agrupar um ou mais PVs.

- *physical extent* (PE): Extensões físicas, ou seja, pequenos pedaços de um PV, um PV é dividido em vários PEs.
- *logical extent* (LE): Extensões lógicas, ou seja, pequenos pedaços de um LV, um LV é dividido em vários LEs

A seguir, tem-se uma visão de como estão organizadas essas unidades:



Além disso, o LVM permite ao administrador escolher entre várias formas de mapeamento dos LE em PV. O tipo de mapeamento pode influenciar no desempenho do sistema de acordo com o

tipo de utilização existente.

Outra facilidade disponível no LVM são os *snapshots* (fotografias) que permitem ao administrador criar cópias exatas de um LV em um determinado momento. Esse processo permite a geração de *backups* (cópias de segurança) ou a remoção de um determinado dispositivo (PV).

3 Utilizando o LVM

Antes de iniciar qualquer procedimento, é recomendável que sejam feitas cópias de segurança dos dispositivos que fizeram parte da instalação. Neste documento, não se trata da instalação do sistema de arquivos raiz (/) utilizando o LVM.

3.1 Pré-requisitos

Os testes realizados nessa documentação foram feitos utilizando-se os pacotes nas versões mostradas na tabela ??.

Pacote	Versão
lvm2	2.00.24-1
dmsetup	1.00.19-2
kernel-image-2.4.27-1-386	2.4.27-6
lvm-common	1.5.17
libdevmapper1.00	2:1.00.19-2

Tabela 1: Versões dos pacotes instalados

Se o seu sistema não tiver essas versões de pacotes instaladas, você poderá enfrentar problemas ou resultados diferentes. Recomenda-se a atualização para essas versões antes de iniciar os procedimentos descritos nessa documentação.

A seguir, mostraremos o procedimento de instalação do pacote LVM.

3.2 Procedimento de instalação

Para instalar o pacote do LVM, deve-se entrar como usuário administrador da máquina (*root*) e após verificar se a configuração do apt está correta, deve-se executar o seguinte comando, respondendo as perguntas que serão feitas:

```
[root@server root]# aptitude install lvm2
```

Observação: No caso da instalação feita para este documento, foi utilizado o lvm2 e, portanto, não foi necessário formatar os discos no padrão Linux LVM. Caso esteja utilizando esse documento para instalar o lvm10, informe-se antes sobre o particionamento dos discos no formato de partição Linux LVM.

Para exemplificar as tarefas, será utilizado um disco de 20GB, cujo dispositivo é o `/dev/hdd`. O disco será dividido - utilizando-se o `fdisk` - em 4 partições de tamanhos iguais (que ocupam todo o disco).

Faça a ativação de um dos modulos que é importante para a utilização do LVM:

```
[root@server root]# modprobe dm-mirror
```

Insira a seguinte linha no arquivo `/etc/modules`, para ativar esse modulo na iniciação do computador:

```
dm-mirror
```

3.3 Iniciando discos ou partições

Para iniciar as partições ou discos, deve-se utilizar o comando `pvcreate`, que cria o descritor de Volume no início do disco ou partição.

Seguindo o exemplo, são iniciadas as 4 partições criadas. Para iniciar a primeira partição, executa-se o seguinte comando:

```
[root@server root]# pvcreate /dev/hdd1
```

E ele deverá gerar a resposta:

```
No physical volume label read from /dev/hdd1  
Physical volume "/dev/hdd1" successfully created
```

E assim sucessivamente:

```
[root@server root]# pvcreate /dev/hdd2  
No physical volume label read from /dev/hdd2  
Physical volume "/dev/hdd2" successfully created
```

```
[root@server root]# pvcreate /dev/hdd3  
No physical volume label read from /dev/hdd3  
Physical volume "/dev/hdd3" successfully created
```

```
[root@server root]# pvcreate /dev/hdd4
No physical volume label read from /dev/hdd4
Physical volume "/dev/hdd4" successfully created
```

Observação: Se estiver iniciando um disco, em vez de uma partição, e esse comando falhar, **apague** a tabela de partições do disco, para isso, execute o comando abaixo. Vale ressaltar que executando este comando, **OS DADOS DO DISCO SERÃO COMPLETAMENTE APAGADOS**.

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

3.4 Visualizando PVs

Utilize o comando *pvdisplay* para saber quais são os PVs disponíveis. No exemplo aqui utilizado, o comando mostra a seguinte saída:

```
[root@server root]# pvdisplay
--- NEW Physical volume ---
PV Name /dev/hdd1
VG Name
PV Size 4,66 GB
Allocatable NO
PE Size (KByte) 0
Total PE 0
Free PE 0
Allocated PE 0
PV UUID RiWHqR-gyYe-jlIQ-pjku-cdI8-cQ8M-Lkj2Y4

--- NEW Physical volume ---
PV Name /dev/hdd2
VG Name
PV Size 4,66 GB
Allocatable NO
PE Size (KByte) 0
Total PE 0
```

```
Free PE 0
Allocated PE 0
PV UUID oEcEui-mhig-RF06-fBd5-opAK-1Psf-bV1FLp

--- NEW Physical volume ---
PV Name /dev/hdd3
VG Name
PV Size 4,66 GB
Allocatable NO
PE Size (KByte) 0
Total PE 0
Free PE 0
Allocated PE 0
PV UUID SwPpNo-Q1NR-OYPU-Tkah-y3ps-GsRs-3W4pSG

--- NEW Physical volume ---
PV Name /dev/hdd4
VG Name
PV Size 4,66 GB
Allocatable NO
PE Size (KByte) 0
Total PE 0
Free PE 0
Allocated PE 0
PV UUID BFGFCi-2rKb-As0v-ucBy-ghkn-y869-yXcIm0
```

3.5 Criando os VGs

Para criar um VG, deve-se usar o comando `vgcreate`. Como foi descrito anteriormente, um VG pode ser visto como um dispositivo. Dessa forma, pode-se agregar vários discos em um “dispositivo virtual”.

Serão criados dois VGs, cada um com 2 discos, que serão nomeados, respectivamente, como VG01 e VG02. Outros nomes, como *arquivos-pessoais* ou *documentos*, por exemplo, também podem ser utilizados.

Para criar o VG01, executa-se o comando:

```
[root@server root]# vgcreate VG01 /dev/hdd1 /dev/hdd2
```

Que gera a resposta:

```
Volume group "VG01" successfully created
```

E para o VG02:

```
[root@server root]# vgcreate VG02 /dev/hdd3 /dev/hdd4
Volume group "VG02" successfully created
```

3.6 Visualizando os VGs

Para ver as informações sobre os VGs existentes, utilize o comando *vgdisplay*.

```
[root@server root]# vgdisplay
--- Volume group ---
VG Name VG02
System ID
Format lvm2
Metadata Areas 2
Metadata Sequence No 1
VG Access read/write
VG Status resizable
MAX LV 255
Cur LV 0
Open LV 0
Max PV 255
Cur PV 2
Act PV 2
VG Size 9,31 GB
PE Size 4,00 MB
Total PE 2384
Alloc PE / Size 0 / 0
Free PE / Size 2384 / 9,31 GB
VG UUID iWiIv6-LOFT-K0gV-JzJh-a8l5-1ouZ-H4r3yN

--- Volume group ---
VG Name VG01
System ID
Format lvm2
```

```

Metadata Areas 2
Metadata Sequence No 1
VG Access read/write
VG Status resizable
MAX LV 255
Cur LV 0
Open LV 0
Max PV 255
Cur PV 2
Act PV 2
VG Size 9,31 GB
PE Size 4,00 MB
Total PE 2384
Alloc PE / Size 0 / 0
Free PE / Size 2384 / 9,31 GB
VG UUID B0o0rf-zv01-uYT0-GW9L-k0ez-H37z-0RzHIh
    
```

As informações que devem ser destacadas na saída desse comando são as seguintes:

- Nome do VG:
VG Name VG02
- Acesso:
VG Access read/write
- Status:
VG Status resizable
- Número Máximo de LVs (Logical Volume):
MAX LV 255
- Número de LVs no VG:
Cur LV 0
- Número de LVs montados:
Open LV 0
- Número Máximo de PVs no VG:
Max PV 255
- Número atual de PVs no VG:
Cur PV 2
- Tamanho do VG em GB/MB:
VG Size 9,31 GB

- Tamanho de cada PE no VG (padrão 4MB):
PE Size 4,00 MB
- Quantidade de PEs no VG. (multiplique este número por 4 para saber o tamanho do VG):
Total PE 2384
- Número de PEs alocados:
Alloc PE / Size 0 / 0
- Número de PEs livres:
Free PE / Size 2384 / 9,31 GB

Use o comando `vgdisplay` com a opção `-v` para ter maiores informações sobre o VG.

3.7 Criando um LV

Para criar um LV, utiliza-se o comando `lvcreate`. Os LVs podem ser visto com “partições” dos VGs. Seguindo o exemplo desse documento, iremos criar o LV01:

```
[root@server root]# lvcreate -l 1000 -n LV01 VG01
Logical volume "LV01" created
```

O comando utilizado criou um LV com 3.91GB. Isso foi possível pois foi criado um LV com 1000 PE, sendo cada um deles com 4MB. Pode-se usar o L maiúsculo como parâmetro do `lvcreate` para determinar o tamanho em MB.

Para obter maiores informações, execute `man lvcreate`.

3.8 Visualizando os LVs

Utilize o comando `lvdisplay`:

```
[root@server root]# lvdisplay
--- Logical volume ---
LV Name /dev/VG01/LV01
VG Name VG01
LV UUID rQw1cG-jppl-TSJj-1wVw-8kud-oE5L-Jr1c46
LV Write Access read/write
LV Status available
# open 0
LV Size 3,91 GB
Current LE 1000
Segments 1
Allocation next free (default)
```

```
Read ahead sectors 0
Block device 253:0
```

O comando `vgdisplay` também mostra informações interessantes sobre os LVs presentes num VG. Veja a saída do comando abaixo:

```
[root@server root]# vgdisplay VG01
--- Volume group ---
VG Name                VG01
System ID
Format                 lvm2
Metadata Areas        2
Metadata Sequence No  2
VG Access              read/write
VG Status              resizable
MAX LV                 255
Cur LV                1
Open LV                0
Max PV                 255
Cur PV                2
Act PV                 2
VG Size                9,31 GB
PE Size                4,00 MB
Total PE               2384
Alloc PE / Size        1000 / 3,91 GB
Free PE / Size         1384 / 5,41 GB
VG UUID                B0o0rf-zv01-uYT0-GW9L-k0ez-H37z-0RzHIh
```

Na saída desse comando, podem ser vistas as seguintes informações:

- Quantidade de LVs disponíveis:
Cur LV 1
- Quantidade de PEs alocados e disponíveis:
Alloc PE / Size 1000 / 3,91 GB Free PE / Size 1384 / 5,41 GB

3.9 Ativando um LV

Como foi dito anteriormente, os LVs podem ser vistos como partições; portanto, para serem utilizados, eles devem ser formatados e montados, como partições comuns.

Nesse exemplo, iremos formatar como sistema de arquivos ext2. A localização dessas “partições” é `/dev/<nome_do_VG>/<nome_do_LV>`, nesse exemplo, vamos formatar a partição LV01 do VG01.

```
[root@server root]# mke2fs /dev/VG01/LV01
mke2fs 1.35 (28-Feb-2004)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
768544 inodes, 1536000 blocks
76800 blocks (5.00%) reserved for the super user
First data block=0
47 block groups
32768 blocks per group, 32768 fragments per group
16352 inodes per group
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912, 819200, 884736

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

This filesystem will be automatically checked every 25 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
```

Pode-se criar qualquer sistema de arquivos que for desejado, pois o sistema operacional considerará os LVs como partições de disco.

Em seguida, cria-se um ponto de montagem para essa partição e se inicia sua montagem, utilizando o /LVM1.

```
[root@server root]# mkdir /LVM1
[root@server root]# mount /dev/VG01/LV01 /LVM1
[root@server root]# df -h
Sist. Arq. Tam Usad Disp Uso% Montado em
/dev/hda2 28G 16G 10G 62% /
/dev/hda1 145M 5,9M 132M 5% /boot
none 507M 0 507M 0% /dev/shm
/dev/mapper/VG01-LV01
3,9G 33M 3,7G 1% /LVM1
```

3.10 Redimensionando LVs

O LV01 foi formatado e ocupa 1000 PEs do VG01. Supondo que se precisa de mais espaço neste LV, pode-se redimensioná-lo, utilizando o comando *lvextend*; que permite tanto o aumento como a diminuição do LV.

Nesse exemplo, o LV01 será aumentado em 500 PEs. Antes, deve-se desmontar a partição.

```
[root@server root]# umount /LVM1
[root@server root]# lvextend -l+500 /dev/VG01/LV01
  Extending logical volume LV01 to 5,86 GB
  Logical volume LV01 successfully resized
[root@server root]# resize2fs /dev/VG01/LV01
  resize2fs 1.35 (28-Feb-2004)
  The filesystem is already 1536000 blocks long. Nothing to do!
[root@server root]# mount /dev/VG01/LV01 /LVM1/
[root@server root]# df -h
  Sist. Arq. Tam Usad Disp Uso% Montado em
  /dev/hda2 28G 16G 10G 62% /
  /dev/hda1 145M 5,9M 132M 5% /boot
  none 507M 0 507M 0% /dev/shm
  /dev/mapper/VG01-LV01
  5,8G 20K 5,5G 1% /LVM1
```

É importante ressaltar que foi necessário redimensionar o sistema de arquivos, para tanto, foi utilizado o comando *resize2fs*¹. Se esse comando falhar e aparecer a seguinte mensagem de erro: `Please run 'e2fsck -f /dev/VG01/LV01' first..` Utilize-o conforme recomendado:

```
[root@server root]# e2fsck -f /dev/VG01/LV01
e2fsck 1.35 (28-Feb-2004)
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/VG01/LV01: 11/512000 files (0.0% non-contiguous), 24287/1024000 blocks
```

Também se pode diminuir o tamanho de um LV, utilizando o comando *lvreduce*.

Observação: O comando *lvreduce* pode causar a perda de dados se não for bem utilizado. Para evitar essa perda de dados, deve-se utilizar o comando *resize2fs*, passando como parâmetro o

¹Para sistemas de arquivos reiserfs, utilize o comando *resize_reiserfs*

tamanho desejado.

```
[root@server root]# umount /LVM1/
[root@server root]# resize2fs /dev/VG01/LV01 2,73G
[root@server root]# lvreduce -l-800 /dev/VG01/LV01
WARNING: Reducing active logical volume to 2,73 GB
THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce LV01? [y/n]: y
Logical volume LV01 successfully resized
[root@server root]# mount /dev/VG01/LV01 /LVM1/
[root@server root]# df -h
Sist. Arq. Tam Usad Disp Uso% Montado em
/dev/hda2 28G 16G 10G 62% /
/dev/hda1 145M 5,9M 132M 5% /boot
none 507M 0 507M 0% /dev/shm

/dev/mapper/VG01-LV01
2,7G 20K 2,6G 1% /LVM1
```

3.11 Apagando um VG

Para apagar um VG, deve-se ter certeza de que ele não possui LVs e deve-se desativar o VG antes de removê-lo. Neste exemplo, o VG02 será apagado:

```
[root@server root]# vgchange -a n VG02
[root@server root]# vgremove VG02
```

3.12 Apagando um LV

Para apagar um LV, deve-se ter certeza de que ele não está montado. Neste exemplo, o LV02 do VG02 será apagado:

```
[root@server root]# umount /dev/VG02/LV02
[root@server root]# lvremove /dev/VG02/LV02
lvremove -- do you really want to remove "/dev/VG02/LV02"? [y/n]: y
lvremove -- doing automatic backup of volume group "LV02"
lvremove -- logical volume "/dev/VG02/LV02" successfully removed
```

3.13 Adicionando PV em um VG

Deve-se utilizar o comando `vgextend` para adicionar PVs que já foram iniciados (como mostrado na seção ??). Por exemplo, para adicionar o disco `/dev/hdc`, no VG02, sendo que esse disco já é um PV (já foi iniciado com o comando `pvcreate`):

```
[root@server root]# vgextend VG02 /dev/hdc
```

3.14 Removendo PV de um VG

Deve-se utilizar o comando `vgreduce` para remover PVs dos VGs. Antes de fazer esse procedimento, deve-se ter certeza de que tal PV não é usado em nenhum LV. Para remover o disco `/dev/hdc` do VG02:

```
[root@server root]# vgreduce VG02 /dev/hdc
```

3.15 Fazendo um Backup de um LV

Para fazer uma cópia de segurança de um LV, é criado de um *snapshot*. Este procedimento cria uma cópia idêntica da LV no momento em que o comando é executado, mesmo que haja mudança do LV durante o procedimento. Ou seja, não é necessário desativar o LV para realizar o backup.

Supondo um LV (chamado de `databases`) de 500 MB num VG (chamado `ops`) que possui mais 500MB de espaço livre. Será criado um novo LV chamado de `dbbackup`.

```
[root@server root]# lvcreate -L500M -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
```

Destacam-se no comando `lvcreate` os seguintes parâmetros:

- `-L500M`:
Define o tamanho do LV criado.
- `-s`:
Definir que um *snapshot* será feito.
- `-n`:
Nome do LV.

O LV é criado com a permissão de *somente-leitura* (*Read-only*), mas pode ser montado como qualquer outro LV. O processo de Backup pode ser feito da maneira que o administrador desejar (fitas, CDs, DVDs, Rsync, etc).

3.16 Retirando um disco antigo

O procedimento descrito nessa sessão é bastante útil quando se descobre que um disco pode ser retirado de um VG, tanto por estar obsoleto, quanto por estar sub-utilizado.

É importante ressaltar que, **sempre se deve fazer um backup dos dados** antes de iniciar esse processo.

Caso haja PEs disponíveis dentro de um VG, de forma a caber todo o conteúdo de um disco que está naquele VG, pode-se simplesmente utilizar o comando *pvmove*, passando como parâmetro o disco que se deseja retirar².

Este comando é bastante lento, pois copia o conteúdo do disco bloco a bloco para os outros discos do VG.

Movendo os dados do `/dev/hdb`, que pertence ao VG chamado `dev`:

```
[root@server root]# 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
```

Agora basta executar o comando *vgreduce*, passando como parâmetro os nomes do VG e do disco que será retirado.

```
[root@server root]# vgreduce dev /dev/hdb
vgreduce -- doing automatic backup of volume group "dev"
vgreduce -- volume group "dev" successfully reduced by physical volume:
vgreduce -- /dev/hdb
```

Caso seja desejável, pode-se utilizar o comando *pvmove* para mover os dados de um disco para outro.

Seguindo o exemplo dado, supondo que o disco `/dev/sdf` foi adicionado ao VG `dev`, basta utilizar o comando:

```
[root@server root]# 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
```

4 Referências bibliográficas

Esse documento foi parcialmente baseado nas seguintes documentações:

²Para obter mais informações, utilize o comando *pvmove* com o parâmetro `-v`.

- Gerenciamento avançado de discos com LVM:
<http://www.dicas-l.unicamp.br/dicas-l/20050122.php>
- LVM HOWTO:
<http://www.tldp.org/HOWTO/LVM-HOWTO/index.html>
- Conversion of the root fs to LVM on woody:
<http://lists.debian.org/debian-user/2002/08/msg00295.html>
- Learning Linux LVM - Part 1:
<http://www-106.ibm.com/developerworks/linux/library/l-lvm/index.html>
- Learning Linux LVM - Part 2:
<http://www-106.ibm.com/developerworks/library/l-lvm2.html>

Recomenda-se a leitura desses artigos para se obter uma melhor compreensão do LVM.

A Créditos

Documento **Utilizando LVM no Debian**

Direitos Autorais Reservados (c)

Universidade Federal de Minas Gerais

Departamento de Ciência da Computação

Laboratório de Software Livre:

Leonardo Luiz Padovani da Mata - barroca@dcc.ufmg.br

Esta documentação é livre; você pode redistribuí-la e/ou modificá-la sob os termos da Licença Pública Geral GNU conforme publicada pela Free Software Foundation; tanto na sua versão 2, como qualquer versão posterior (a seu critério) .

A distribuição desta documentação é feita na expectativa de que ela seja útil, porém, **sem nenhuma garantia**; nem mesmo a garantia implícita de **comerciabilidade ou adequação a uma finalidade específica**.

Consulte a Licença Pública Geral do GNU para mais detalhes.



<http://creativecommons.org/licenses/GPL/2.0/>

<http://creativecommons.org/licenses/GPL/2.0/legalcode.pt>