

Deise Springer de Freitas

Utilização do LVM como Facilitador de Gerencia de Partições no Linux

Monografia de Pós-Graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências da disciplina Monografia para obtenção do título de Especialista em Administração de Redes Linux.

Orientador
Prof. Dr. Gustavo Guimarães Parma

Lavras
Minas Gerais - Brasil
2005

Deise Springer de Freitas

**Utilização do LVM como Facilitador de Gerencia de
Partições no Linux**

Monografia apresentada ao Departamento de
Ciência da Computação da Universidade Federal
de Lavras como parte das exigências do curso de
PósGraduação *Lato Sensu* Administração em
Redes Linux, para obtenção de título de
especialista.

Aprovada em 17 de Abril de 2005

Prof. Sandro Melo

Prof. Heitor Augustus Xavier Costa

Prof. Dr. Gustavo Guimarães Parma
(Orientador)

LAVRAS
MINAS GERAIS BRASIL

Dedico este trabalho a meus pais e aos
meus filhos Fábio e Lia.

Agradecimentos

Agradeço a todos que me ajudaram, especialmente a meu orientador prof. Dr. Gustavo Parma, ao Prof. Msc. Joaquim Quinteiro Uchôa e aos meus colegas do Banco Central.

Resumo

Quando um novo servidor é instalado pode ser difícil definir como será feito o particionamento do(s) disco(s), pois é difícil prever quanto de espaço será necessário disponibilizar para usuários, para sistemas ou para aplicações. Por mais que se conheça o ambiente, é comum que uma das partições acabe enchendo e outra(s) fique(m) com espaço sobrando, sendo necessário remanejar o espaço entre as partições. Além disto, a distribuição inicial dos dados pode ser feita de forma que alguns discos fiquem com excesso de acessos e outros praticamente sem uso, o que exige o remanejamento dos dados para melhoria de desempenho do servidor. Neste trabalho, são apresentados os principais conceitos teóricos e um exemplo de implementação do LVM, mostrando como ele trata estes problemas e facilita o gerenciamento de espaço em disco.

Sumário

1 - Introdução.....	1
2 - Partições de Disco.....	3
2.1 Ferramentas Para Particionamento de Disco.....	5
2.1.1 - Fdisk	5
2.1.2 – Cfdisk.....	5
2.1.3 – Disk Drake.....	6
2.1.4 – Fips.....	6
2.1.5 - Partition Magic.....	6
2.1.6 - Acronis Partitionexpert 2003.....	7
2.1.7 - GNU Parted (GNU partition editor).....	7
3 - Logical Volume Manager (LVM)	8
3.1 – Conceitos.....	8
3.1.1 -Volume Físico (PV).....	9
3.1.2 -Grupo de Volume (VG).....	9
3.1.3 – Extensão Física (PE).....	9
3.1.4 – Volume Lógico (LV).....	10
3.1.5 – Extensões Lógicas (LE).....	12
3.1.6 – Snapshot.....	13
3.2 - Vantagens do LVM.....	13
3.3 - Desvantagens do LVM.....	15
4 – Implementação.....	16
4.1 – Ambiente Utilizado.....	16
4.2 – Criação dos VGS	16
4.3 – Criação dos LVS	18
4.4 – Formatação dos sistemas de arquivos.....	20
4.5 - Uso de snapshots.....	20
4.6 - Redimensionamento de um LV.....	21
4.7 - Adição de um novo disco ao VG.....	23
4.8 – Movimentação de PVS.....	24
4.9 – Movimentação um VG para outro servidor.....	24
4.10 – Cópia de Segurança da Configuração do LVM.....	26
5 - CONCLUSÃO.....	29
Referências Bibliográficas	31

LISTA DE FIGURAS

Figura 1 – Exemplo de VG, PV e PE.....	10
Figura 2 - Volumes Lógicos (LV) e grupos de volumes (VG).....	11
Figura 3 – Formação de LV por stripe ou concatenação.....	12
Figura 4 – Mapeamento entre LE e PE.....	12
Figura 5 – Diagrama da implementação – criação dos VGs.....	17

1 - Introdução

A tarefa de administrar servidores Linux envolve um grande número de atividades e, sem dúvida, um dos grandes desafios é a administração do espaço de armazenamento.

Uma definição difícil para quem está instalando o Linux é a forma como será feito o particionamento do(s) disco(s), pois é difícil prever quanto de espaço será necessário disponibilizar para usuários, para sistemas ou para aplicações. Por mais que se conheça o ambiente, é comum que uma das partições acabe enchendo e outra(s) fique(m) com espaço sobrando, sendo necessário remanejar o espaço entre as partições.

Este trabalho foi elaborado devido à necessidade de se encontrar ferramentas para facilitar a administração de servidores Linux usados para gerenciadores de bancos de dados e com volume de armazenamento superior a 1,5 terabytes. O seu objetivo é buscar uma forma mais fácil de gerenciar o espaço em disco, principalmente em ambientes em que a área de armazenamento é muito grande e o crescimento de utilização é imprevisível, como no caso descrito.

Como solução para estes tipos de problema, este trabalho faz uma comparação entre diversas ferramentas de particionamento e propõem a utilização de volumes lógicos (LVM – *Logical Volume Manager*) nos servidores para facilitar a administração do espaço em disco.

A monografia está organizada como apresentado a seguir: no Capítulo 2, são apresentadas as principais formas de particionamento de disco. No Capítulo 3, são vistos os conceitos usados pelo LVM e as vantagens e desvantagens do seu uso. No Capítulo 4, é apresentado um exemplo de implementação, em que foram criados VGs (*Volumes Groups*) e LVs (*Logical volumes*) de formas diferentes, sendo mostradas as seguintes funcionalidades do

LVM: uso de *snapshots*, redimensionamento de volumes, adição e remoção de discos, cópia de volumes lógicos, movimentação de volumes físicos e transferência de VG para outros servidores. No Capítulo 5, são apresentadas as conclusões sobre este trabalho. Por fim, são apresentadas as Referências Bibliográficas utilizadas.

2 - Partições de Disco

Os discos rígidos, quando são fabricados, não estão preparados para receber dados. Para isso, eles precisam ser formatados. Esta formatação é aplicada em uma partição, onde uma partição representa uma determinada área física da mídia do disco designada para ser utilizada. É necessário que o disco seja particionado antes da formatação. Existem dois tipos de formatação: formatação física ou de baixo nível, em que são criadas as trilhas e setores do disco, e formatação lógica ou de alto nível, em que é criado o sistema de arquivos. Sistema de arquivos diz respeito à estrutura de dados criada no disco rígido que permite a remoção, a criação e a cópia de arquivos, dentre outras funcionalidades. O Linux suporta vários sistemas de arquivos, sendo os principais, atualmente, *ext2*, *ext3*, *reiserfs* e *swap* [3].

As partições criadas no disco podem ter, teoricamente, qualquer tamanho. O limite de cada partição depende do sistema de arquivos a ser utilizado na partição. Por exemplo, em uma partição utilizando *ext2*, o maior tamanho da partição será 4 TB [6]. Porém, uma partição usando FAT32 terá o tamanho máximo de 32 GB [9]. Caso seja criada uma partição única, todo o disco rígido é usado para armazenar os arquivos. Se forem criadas várias partições, estas podem ser usadas para fins distintos, uma partição independente das demais. Por exemplo, cada partição pode conter um sistema operacional. Um exemplo de aplicação de várias partições, é a presença de duas distribuições GNU/Linux no mesmo disco. Os dois sistemas, apesar de estarem no mesmo disco, não interferem um no outro.

Os seguintes dados sobre todas as partições existentes no disco são armazenados no disco rígido, na tabela de partições: sistema de arquivos utilizado, se ela é inicializável, a cabeça inicial/final, o setor inicial/final, o

cilindro inicial/final e o número de setores ocupados no disco.

Existem três tipos possíveis de partições: primária, estendida e lógica.

As partições primárias permitem o armazenamento de arquivos e diretórios. Em um disco, deve haver no mínimo uma e no máximo quatro partições primárias. Se existirem quatro partições primárias, nenhuma outra partição poderá existir neste disco, nem de outro tipo. Uma dessas partições deve, necessariamente, estar marcada como ativa e “bootável” (ou inicializável) para que a BIOS possa iniciar a máquina por ela.

Só pode haver uma partição estendida em cada disco. Uma partição estendida é um tipo especial de partição primária utilizada para a criação de outras partições dentro dela, chamadas de partições lógicas. Se existir uma partição estendida, ela toma o lugar de uma das partições primárias. Desta forma, havendo uma partição estendida no disco poderão haver, no máximo, outras três partições primárias [16].

A partição estendida pode comportar até 64 partições lógicas [12]. Desta forma, um mesmo disco rígido pode armazenar mais de 64 partições independentes uma das outras.

Existem vários motivos para dividir um disco rígido em partições múltiplas:

1. Organizar o ambiente: manter separados sistemas operacionais, aplicações e dados. Dessa forma, pode-se instalar vários sistemas operacionais em partições diferentes, colocar os dados em uma partição separada para facilitar o *backup* e a instalação de novos sistemas operacionais;
2. Otimização: recuperar o espaço perdido por *clusters* ineficientes, utilizando partições menores. Ter diversos sistemas de arquivos em partições diferentes. Além disto, pode-se usar diferentes tamanhos de blocos para aplicações diferentes - para aplicações com arquivos grandes, alocação de blocos muito pequenos implica em perda de desempenho. Por outro lado, vários arquivos pequenos gravados em blocos grandes provocam perda de espaço em disco;

3. Segurança: separar dados das aplicações e do sistema operacional. Colocar dados sigilosos em partições que podem ser escondidas por utilitários. Além disto, normalmente a corrupção do sistema de arquivo se limita a uma partição, de modo que apenas parte dos dados são perdidos se ocorrer algum acidente [4].

2.1 Ferramentas Para Particionamento de Disco

No Linux, há várias ferramentas para particionamento de discos, sendo que as mais comuns, segundo [12], são: o fdisk e o cfdisk (com interface de menus em modo texto).

Todas estas ferramentas trabalham com o particionamento de um disco de cada vez, não permitindo a criação de partições com tamanho maior que um disco físico, a menos que seja usada outra ferramenta (LVM ou Raid lógico, por exemplo) para criar discos lógicos reconhecidos pelo sistema operacional como um disco físico.

2.1.1 - Fdisk

É a ferramenta mais básica de particionamento, não devendo ser confundida com o produto homônimo do DOS [16], pois é totalmente diferente. É uma ferramenta de linha de comando, que é usado de forma interativa, ou seja, espera por comandos do usuário [14].

2.1.2 – Cfdisk

É um *front-end* para o fdisk, baseado em *curses* (interface texto com menus navegáveis). Nesta ferramenta existem as mesmas opções do fdisk só que com uma interface mais agradável [16].

2.1.3 – Disk Drake

Programa Perl/Gtk para criar, redimensionar, redefinir e remover partições. Ao redimensionar as partições, há perda de dados, exceto para partições FAT [2].

2.1.4 – Fips

FIPS (*First Interactive Partition Splitter*) - Utilitário presente em todas grandes distribuições. Apesar de rodar em DOS e necessitar da digitação de comandos no *prompt*, não é difícil de usar. É eficiente e possibilita salvar a tabela de partições antes de alterá-la, de modo que em caso de algum erro seja possível voltar para o estado anterior [12].

2.1.5 - Partition Magic

Aplicação comercial usada para criar, copiar, redimensionar, fazer merge (união) e converter partições sem perda de dados. O tamanho das partições se limita a 300 GB. O Partition Magic só pode ser instalado em ambiente Windows, de modo que para ser usado com GNU/Linux é necessário parar o equipamento e inicializar o sistema usando um disquete com a versão DOS ou um cd com o *BootMagic configuration program*. A vantagem desta aplicação é a facilidade de uso. Mais informações sobre o Partition Magic

podem ser obtidas em [10].

2.1.6 - Acronis Partitionexpert 2003

Aplicação comercial usada para criar, redimensionar, copiar e converter partições sem perda de dados. O espaço alocado para cada partição precisa ser contíguo, de modo que, para aumentar uma partição é necessário que haja espaço livre disponível imediatamente antes ou após a mesma. Existem versões tanto para Linux quanto para Windows. Mais detalhes sobre o produto podem ser vistos em [11]

2.1.7 - GNU Parted (GNU partition editor)

É um programa desenvolvido e mantido pelo australiano Andrew Clausen para criar, destruir, redimensionar, verificar e copiar as partições e os sistemas de arquivos destas partições sem perdas de dados. O endereço inicial da partição não pode ser alterado, de forma que para aumentá-la deve ser alterado o endereço final. O espaço alocado para cada partição precisa ser contíguo, de modo que, para aumentar uma partição é necessário que haja espaço livre disponível logo após esta partição. É útil para liberar espaço para novos sistemas operacionais, reorganizar o uso do disco, copiar dados entre discos rígidos e fazer imagens de discos. O programa pode funcionar interativamente com o fdisk ou em linha de comando. O Parted possui a mesma funcionalidade que o Partition Magic, com a vantagem de ser software livre e GPL. É recomendável a leitura de [1] para mais informações sobre este utilitário.

3 - Logical Volume Manager (LVM)

O LVM foi desenvolvido originalmente pela IBM e adotado pelo OSF (antiga *Open Software Foundation* e atual *Open Group*). A versão OSF foi usada como base para implementação do LVM nos sistemas operacionais HP-UX e Digital UNIX. A implementação do Linux é similar à implementação de LVM do HP-UX.[15]. O projeto LVM começou em fevereiro de 1997 e a versão 0.1 foi disponibilizada em julho de 1997. A versão 1.0 foi lançada em agosto do mesmo ano [8]. A última versão, disponível em <ftp://sources.redhat.com/pub/lvm2/>, é a 2.01.04.

O LVM cria uma camada de abstração do sistema de armazenamento para as aplicações diferente da visão tradicional de discos e partições. Este armazenamento em volumes lógicos disponibiliza um meio, invisível para usuários e aplicativos, de dividir e alocar o espaço dos discos. Quando se usa o LVM, os discos físicos são divididos em unidades de armazenamento. Essas unidades de armazenamento, de diversos discos, podem ser agrupadas em um volume lógico, que pode ser dividido em “partições”. A qualquer momento, novas unidades de armazenamento podem ser adicionadas ou removidas das partições[7].

3.1 – Conceitos

O LVM utiliza diversos conceitos que precisam ser entendidos para não haver problemas em utilizá-lo. Nesta subseção será apresentada a terminologia utilizada pelo LVM.

3.1.1 -Volume Físico (PV)

Para ser utilizado pelo sistema, o espaço disponível em disco deve passar por um processo de configuração, o qual prepara o disco para o trabalho com o LVM. Todo disco rígido ou partição, para trabalhar com o LVM, precisa se tornar um *Physical Volume* (PV - volume físico). Dispositivos que tradicionalmente são vistos pelo sistema operacional como discos físicos, como por exemplo os sistemas raid¹ ou partições de disco IDE, também são considerados como PV.

Ao criar um PV, é definida, no início do disco físico, a tabela de descrição do grupo de volumes (VGDA – *Volume Group Descriptor Area*). Segundo [17], o VGDA contém as seguintes informações: um descritor do PV, um descritor do VG, os descritores dos LV e dos *Physical Extensions*.

3.1.2 -Grupo de Volume (VG)

Os PVs são agrupados em pools chamados *Volume Groups* (VG - grupo de volumes). Cada VG pode ser formado por uma ou mais partições de tamanhos variados, mas é sempre visto pelo sistema como se fosse um único disco físico. Sendo assim, é possível adicionar ou remover volumes físicos aos VGs sem a necessidade de parar o sistema.

3.1.3 – Extensão Física (PE)

Ao adicionar um PV a um VG, o PV é dividido em partes contínuas e de

¹ Raid = **Redundant Array of Independent Disks**, *Arranjo redundante de discos*, é um meio de criar uma unidade virtual composta por vários discos físicos, com a finalidade de duplicação (redundância) ou balanceamento de operações de entrada/saída.

tamanhos idênticos, chamadas Physical Extensions (PE ou extensões físicas). A PE é a menor unidade que pode ser endereçada pelo LVM nos dispositivos físicos. Em cada PV, o número de identificação de cada PE é único e numerado de 0 (zero) até o número de PEs (tamanho do disco dividido pelo tamanho de cada PE) [5].

A Figura 1 permite visualizar os conceitos de PE, PV e VG:

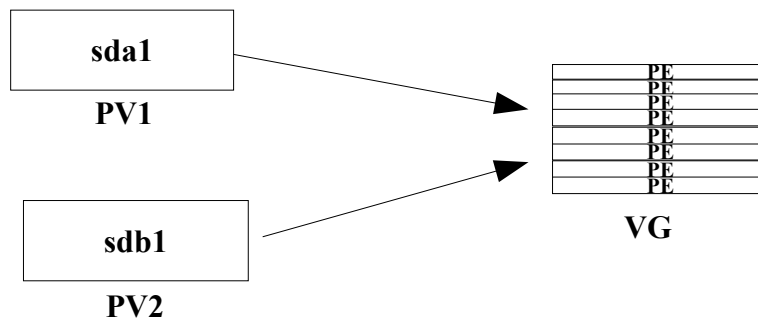


Figura 1 – Exemplo de VG, PV e PE

Nessa figura, dois discos físicos (sda1 e sdb1) são configurados como PVs (pv1 e pv2). Estes PVs se juntam para formar um VG, de forma que os dois PVs são divididos em várias extensões físicas (PE) do mesmo tamanho.

3.1.4 – Volume Lógico (LV)

Não é possível criar ou montar sistemas de arquivos (FS ou *file systems*) ou partições *raw*² diretamente nos VGs. Para isto, é necessário criar um ou mais volumes lógicos (LV ou *logical volumes*) em cada VG [13]. Cada LV é considerado como uma partição para o sistema. A Figura 2, retirada de [5], ilustra o conceito de LV e mostra como os volumes lógicos podem ser definidos em um VG. Ela mostra a concatenação dos discos físicos sda e sdb, de 18 GB

² Partição *raw* é aquela não gerenciada pelo kernel, mas com acesso de baixo nível para aplicações, geralmente usadas por gerenciadores de banco de dados, como Oracle ou DB2 UDB.

cada, para formar o VG identificado por vg1, de 36 GB. Neste VG, são criados 3 volumes lógicos: lv1, lv2 e lv3. São nesses LVs que os sistemas de arquivo são montados.

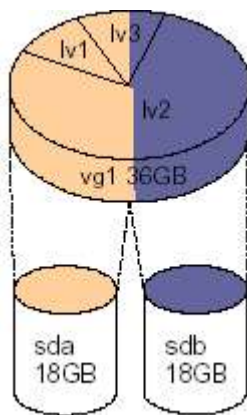


Figura 2 - Volumes Lógicos (LV) e grupos de volumes (VG)

Os LVs podem ser formados de duas formas: por concatenação de PVs, ou formando *stripes*, como pode ser visto na Figura 3. Um *stripe* é formado distribuindo os dados em vários discos. A vantagem do mapeamento dos dados em *stripe* é o melhor desempenho, devido ao paralelismo de acesso aos dados. Em compensação, a utilização de *stripes* sem medidas adicionais aumenta as chances de falhas, pois se houver falha em um disco, todo o Volume Lógico irá falhar junto. Se os dados forem concatenados, apenas uma parte do sistema de arquivos será perdida [15].

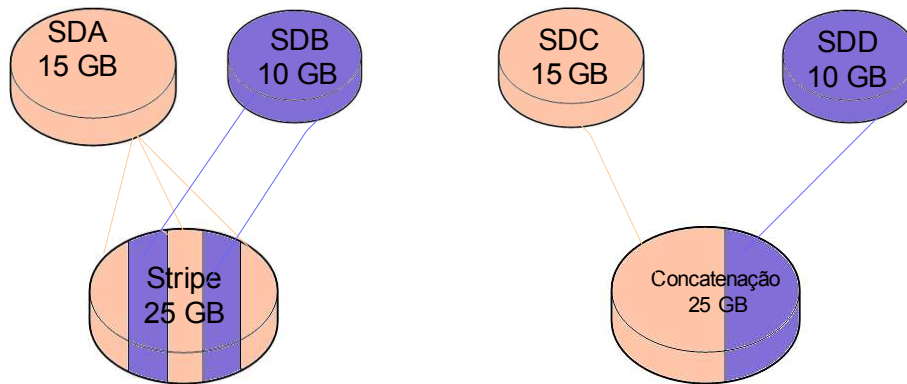


Figura 3 – Formação de LV por *stripe* ou concatenação

3.1.5 – Extensões Lógicas (LE)

Cada LV é dividido em partes contínuas e de tamanhos idênticos, chamadas *Logical Extensions* (LE ou extensões lógicas). Cada LE é mapeada para exatamente uma PE no VG. O número que identifica cada LE é único por VG, da mesma forma que o identificador de cada PE é único por PV, como mostra a Figura 4, retirada de [5].

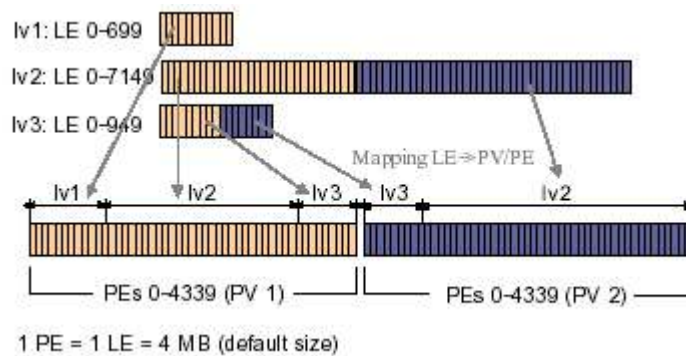


Figura 4 – Mapeamento entre LE e PE

Quando uma aplicação acessa um disco usando LVM, o endereço do LE no LV é utilizado para determinar a posição física no disco, através da tabela de mapeamento. Por exemplo, considerando a Figura 4, verifica-se que o disco físico 1 (PV1) contém os 700 LEs do lv1, parte de lv2 e aproximadamente metade de lv3.

3.1.6 – Snapshot

Uma facilidade adicional do LVM são os *snapshots*. Eles permitem que se crie um novo dispositivo de bloco que é a cópia exata de um LV em um determinado momento e pode ser montado apenas para leitura. Normalmente, essa facilidade é usada por algum processo em lote, como a realização de backup, que necessita usar o LV por algum período sem que os dados sejam alterados durante a utilização. Quando o processo termina, o *snapshot* pode ser removido. É necessário que o *snapshot* seja realizado em um instante em que os dados estejam consistentes.

3.2 - Vantagens do LVM

Uma tarefa difícil para quem está instalando o Linux pela primeira vez é decidir como será feito o particionamento do(s) disco(s), pois é difícil prever quanto de espaço será necessário disponibilizar para usuários, para sistemas ou para aplicações. Quando uma das partições enche e outra(s) fica(m) com espaço sobrando, é necessário remanejar o espaço entre as partições. Com o uso do LVM, o remanejamento de espaço entre partições, que passam a ser LV, fica fácil e pode ser feita sem reinicialização do equipamento, de forma transparente para as aplicações. Uma boa opção, com o LVM, é alocar apenas o espaço

necessário para cada partição e deixar espaço livre nos VGs, expandindo os LVs à medida que for necessário.

Com o uso do LVM, as aplicações passam a usar discos virtuais, não precisando mais se preocupar com a localização física dos dados, em que tipo ou em que lugar do discos eles estão. Separando a gerência dos dispositivos físicos da gerência de armazenamento (*software*), é possível fazer mudanças de *hardware* sem que as aplicações sejam afetadas, mesmo sem pará-las. Sendo assim, a utilização do LVM aumenta o tempo sem paradas programadas do sistema, ao evitar que mudanças na configuração do sistema de armazenamento requeiram paradas do sistema.

Além disso, a adição de discos torna-se tarefa fácil, pois basta adicionar discos a VGs pré-existentes e aumentar os LVs, sem necessidade de migrar dados ou parar o sistema. Esta facilidade é particularmente útil quando os equipamentos usados permitem que a adição ou remoção de discos físicos seja feita sem a necessidade de parar o sistema (um subsistema de discos externo, por exemplo).

A movimentação de dados entre discos também pode ser efetuada usando o LVM. É possível mover todo um LV de um disco para outro, sem afetar a disponibilidade das aplicações.

Outra vantagem é na definição de volumes virtuais (VG) e volumes lógicos (LV), a possibilidade de usar nomes significativos, como *desen*, *pessoal*, *contab*, *teste*, em vez dos nomes de discos e partições físicas, como *sda* e *sda1*. Em ambientes corporativos, com grande quantidade de discos, esta facilidade torna o gerenciamento do sistema de armazenamento menos complexo.

A possibilidade de criar LVs em modo *stripe* permite a criação de volumes lógicos com melhor desempenho, devido ao espalhamento dos dados em mais de um volume.

A utilização de *snapshots* para realização de *backups* é particularmente útil em servidores de arquivos, pois pode-se manter o *snapshot* das últimas 24h

ou mais, permitindo a recuperação rápida de arquivos removidos ou alterados por engano.

3.3 - Desvantagens do LVM

A camada LVM no kernel mantém um mapeamento entre as localizações lógicas e físicas dos discos. Desta forma, para cada leitura ou escrita em disco é necessário que o LVM encontre a posição física da área desejada em uma tabela. Assim, há um gasto de processamento extra, porém pequeno, visto que a busca em uma tabela de mapeamento é uma das operações mais rápidas e básicas efetuadas pelo computador [5].

Por ser uma ferramenta sofisticada, é necessário tempo para aprender a usar o LVM, além de ser necessário maior esforço de manutenção. Antes de usar o LVM é necessário conhecer bem como ele funciona e todos os conceitos envolvidos.

É necessário manter sempre um backup atualizado da configuração do LVM, de forma a evitar a perda de dados no caso das informações de configuração serem corrompidas. Além disto, se forem usados LVs com dados em mais de um volume físico, a perda de um disco acarretará a perda de todo o LV.

Sendo assim, em ambientes estáveis, em que o crescimento de cada partição é fácil de ser determinado com pequena possibilidade de erro, o uso de LVM não é recomendável, pois nesse caso ele não agregará facilidades na administração.

4 – Implementação

Com o objetivo de tornar mais claros os conceitos escritos anteriormente e o modo de utilização do LVM, é descrito um exemplo de implementação e uso das facilidades disponíveis.

4.1 – Ambiente Utilizado

O ambiente utilizado para este exemplo é composto de um servidor que dispõem inicialmente de 4 discos de um array externo, sendo que três deles têm 72 GB disponíveis e o quarto 14 GB para os dados: /dev/sdb, /dev/sdc, /dev/sdd e /dev/sde, respectivamente.

São necessários os seguintes recursos para usar o LVM2: o device-mapper precisa estar disponível no kernel, a biblioteca de suporte ao userspace device-mapper (libdevmapper) e as ferramentas de userspace LVM2.

Para executar este exemplo, foi utilizado o Fedora Core 2. Ao instalar esta distribuição, o LVM versão 2 é disponibilizado automaticamente, sendo possível usar o LVM para todas as partições, inclusive a de *boot*.

4.2 – Criação dos VGS

O primeiro passo para a criação de um VG é a criação dos PVs. É usual criar partições nos discos antes de formatar o disco como PV, mesmo que estas partições usem o disco inteiro, apenas objetivando a documentação. No LVM1, era necessário criar partições do tipo 0x8e, porém com o LVM2 isto não é mais necessário. Este tipo de partição ainda é usado apenas para facilitar a administração dos volumes, pois se não houver esta formatação não é possível identificar quais partições estão sendo usadas por LVM quando se usa o fdisk.

Para o exemplo que será descrito, foram criadas inicialmente duas partições em cada disco, com tamanhos diferentes, sendo formados dois VGs com partições dos três discos, como mostra a Figura 5.

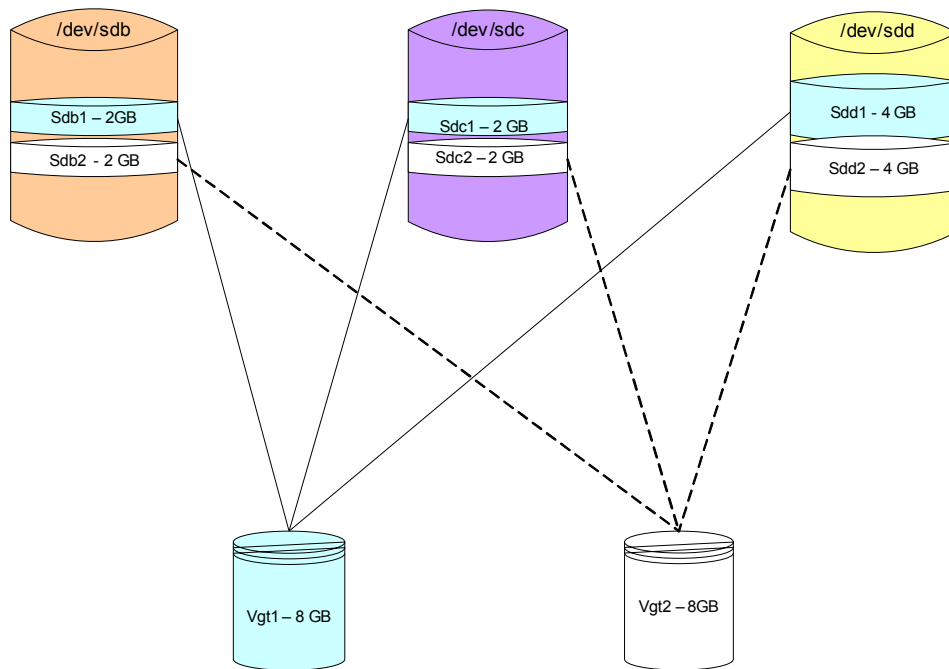


Figura 5 – Diagrama da implementação – criação dos VGs

Para criar estes Vgs, foram seguidos os seguintes procedimentos:

- 1 Criação de 2 partições de 2 GB em cada um dos discos /dev/sdb e /dev/sdc e 2 partições de 4 GB no /dev/sdd. Às seis partições, foi atribuído o tipo “8e”, utilizando-se o fdisk.
- 2 Formatação dos Pvs, através do comando pvcreate como no exemplo abaixo, que foi executado para todas as partições criadas:

```
# pvcreate /dev/sdb1
```

O comando `pvccreate` cria uma área de descrição do volume (VGDA) no início da partição

3 Definição dos VGs nos discos `/dev/sdb`, `/dev/sdc` e `/dev/sdd`, usando o comando `vgcreate`. Neste comando, se não for especificado o tamanho de cada PE o valor de 4KB é assumido, o que limita o tamanho dos LV criados no VG a 256 GB, pois cada LV pode ter no máximo 64 PEs. A fim de aumentar este limite, foram usados PEs de 64 KB, o que teoricamente elevaria este limite a 4 TB (64 x 64KB), porém devido a limitações do sistema operacional o tamanho máximo de cada LV é 2 TB [man `vgcreate`].

Os VGs foram criados usando os seguinte comandos :

```
# vgcreate -s 64 vgt1 /dev/sdb1 /dev/sdc1 /dev/sdd1
```

```
# vgcreate -s 64 vgt2 /dev/sdb2 /dev/sdc2 /dev/sdd2
```

O primeiro comando `vgcreate` criou um novo `vg` com o nome `vgt1`, usando as partições `/dev/sdb1`, `/dev/sdc1` e `/dev/sdd1`. O parâmetro `-s` define o tamanho de cada PE, que no caso é de 64 KB.

O disco `/dev/sde` não foi utilizado na criação dos Vgs porque ficou reservado para testar a adição de um novo disco a um VG.

4.3 – Criação dos LVS

Foram criados dois LVs em cada VG, sendo o primeiro por concatenação e o segundo formando stripe, cada um de 1 GB. O comando utilizado para a criação dos LVs é o `lvcreate`.

Para criar o primeiro LV usando concatenação, foi utilizado o seguinte comando:

```
# lvcreate -L1G -nlvl vgt1
```

Nesse comando, o parâmetro `-L` define o tamanho do LV que está sendo criado, 1 GB no caso. O novo LV foi criado no VG `vgt1` e seu nome é

lv1, informado pela opção -n.

Observou-se, usando o comando pvdisplay, que o LV criado ocupou os PEs 00 a 15 do disco /dev/sdb1 e os LEs 00 a 15 no mesmo disco. Os outros discos permaneceram vazios após a criação deste LV.

Para criar o segundo LV usando stripe, foi utilizado o seguinte comando:

```
# lvcreate -i3 -l32 -L1G -nlvs vgt1
```

O parâmetro -i3 indica que o stripe deverá ser feito entre os três PVs disponíveis e l32 informa que o tamanho de cada pedaço do stripe deverá ser de 32 Kb. Este tamanho foi escolhido por ser o mesmo tamanho usado para leitura e gravação pelo gerenciador de banco de dados.

Observou-se, usando o comando pvdisplay, que o LV criado ocupou os PEs 16 a 21 do disco /dev/sdb1 e os PEs 00 a 05 dos discos /dev/sdc1 e /dev/sdd1. Foram ocupados por este LV os LEs 00 a 05 no /dev/sdb1, 06 a 11 no /dev/sdc1 e 12 a 17 no /dev/sdd1.

A análise da tabela 1 permite verificar a diferença de alocação de espaço em um LV criado por concatenação (lv1) e um criado por stripe (lvs).

	PE			LE		
	Sdb	Sdc	Sdd	Sdb	Sdb	Sdd
Lv1	00 a 15	-	-	00 a 15	-	-
Lvs	16 a 21	00 a 05	00 a 05	00 a 05	06 a 11	12 a 17

Tabela 1 – Diferença de distribuição de espaço entre LV striped e concatenado

Apesar do tamanho especificado para os dois LVs ter sido o mesmo, 1 GB, o lvs (striped) ficou com o tamanho um pouco maior, ocupando 18 Pes, enquanto o lv1 (concatenado) ocupou 16. Isto ocorreu porque o tamanho de 1 GB não daria uma quantidade equivalente de PEs nos três PVs utilizados e o lvcreate arredondou para o próximo tamanho que permitisse essa equivalência.

4.4 – Formatação dos sistemas de arquivos

A fim de utilizar os LVs criados sem um gerenciador de banco de dados, é necessário usar um sistema de arquivos. Para verificar as diferenças de utilização do LVM com sistemas de arquivo diferentes, foram feitos testes usando dois sistemas de arquivo: ext3 e reiserfs. Estes sistemas de arquivo foram usados como exemplo devido ao fato de, segundo [3], serem os tipos mais usados em Linux atualmente. Os LVs criados em vgt1 foram formatados com o sistema de arquivos ext3 usando o comando `mke2fs`. Os LVs criados em vgt2 foram formatados com o sistema de arquivos reiserfs, usando o comando `mkfs.reiserfs`.

4.5 - Uso de *snapshots*

Antes de criar o snapshot foram criados os arquivos teste1 e teste2 no filesystem montado em `/dev/vgt1/lvs`. O conteúdo do arquivo teste2 era “Teste de snapshot – original”. Para criar um snapshot (parâmetro `-s` do comando `lvcreate`) do LV `/dev/lvt1/lvs` com o tamanho de 200 Mb e nome `lvsnaps`, foi usado o seguinte comando:

```
# lvcreate -L200M -s -nlvsnaps /dev/vgt1/lvs
```

Ao usar o comando `lvscan` para verificar o snapshot criado, foi obtido o seguinte resultado:

```
# lvscan
lvscan -- ACTIVE      Original  "/dev/vgt1/lvs" [1.12 GB] striped[3]
lvscan -- ACTIVE      "/dev/vgt1/lvl" [1 GB]
lvscan -- ACTIVE      Snapshot "/dev/vgt1/lvsnaps" [255.75 MB] of /
dev/vgt1/lvs
lvscan -- 3 logical volumes with 2.37 GB total in 2 volume groups
lvscan -- 3 active logical volumes
```

Depois disto, o LV de snapshot foi montado com o comando:

```
#mount /dev/vgt1/lvsnap /mnt/snap
```

O volume de snapshot é montado apenas para leitura, não sendo possível alterá-lo. O volume criado é a cópia exata de /stripe. Observou-se que, apesar do LV de snapshot ter apenas 255,75 MB, ao utilizar o comando df, o seu tamanho aparece igual ao tamanho do LV original, como se pode ver na saída abaixo:

```
/dev/vgt1/lvs      1161104  22780 1079344  3% /stripe  
/dev/vgt1/lvsnap  1161104  22780 1079344  3% /mnt/snap
```

O arquivo teste1 foi apagado, o conteúdo de teste2 foi alterado para “teste de snapshot – alterado” e foi criado o arquivo teste3 no diretório /stripe. Observou-se que não há alteração em /mnt/snap, ou seja, o arquivo teste1 continua no diretório, o conteúdo de teste2 é “Teste de snapshot – original” e não existe o arquivo teste3 em /mnt/snap. Conclui-se que o conteúdo do volume de snapshot é realmente uma imagem do volume no momento que se fez a cópia, não sendo modificado por alterações posteriores no volume original. Esta imagem é muito útil para o uso em backup, pois permite que o usuário permaneça usando o diretório original enquanto a cópia é feita a partir do volume de snapshot. Desta forma, garante-se um backup íntegro sem afetar a disponibilidade do sistema.

Para remover o volume de snapshot, basta desmontá-lo e usar o comando lvremove:

```
#umount /dev/vgt1/lvsnap  
#lvremove /dev/vgt1/lvsnap
```

4.6 - Redimensionamento de um LV

Para aumentar ou diminuir o tamanho de um LV, mesmo com ele montado, basta usar o comando `lvextend`. Porém, para que o novo espaço fique disponível para uso, é necessário que o sistema de arquivos utilizado também permita fazer esta alteração de tamanho online. No caso do Linux, o sistema de arquivos mais usado é o `ext3`, porém, usando o Fedora Core 2, é necessário desmontar a partição com `ext3` para depois aumentar o tamanho. No caso do `reiserfs`, esta alteração pode ser feita online. Sendo assim, foram feitos dois testes, um usando `ext3` e outro usando o `reiserfs`.

Inicialmente os LVs `/dev/vgt1/lv1` e `/dev/vgt1/lvs`, que foram formatados com o sistema de arquivos `ext3`, foram aumentados usando os seguintes comandos para cada lv:

```
#umount /dev/vgt1/lv1
#e2fsadm -L+1G /dev/vgt1/lv1
#mount /dev/vgt1/lv1
#umount /dev/vgt1/lvs
#e2fsadm -L+1G /dev/vgt1/lvs
#mount /dev/vgt1/lvs
```

O comando `e2fsadm` permite que seja feito o redimensionamento de volumes lógicos contendo sistemas de arquivo `ext2`. Para fazer isto com o sistema de arquivos montado, é necessário que a distribuição Linux utilizada disponibilize a aplicação `ext2online`, o que não ocorre com o Fedora, mesmo na última versão. Para sistemas de arquivos desmontados, o `e2fsadm` estende ou diminui o volume lógico e o sistema de arquivos. O parâmetro `-L` redimensiona o sistema de arquivos para o tamanho especificado. Se for especificado `+` ou `-`, o tamanho é em relação ao espaço alocado, senão o volume fica do tamanho especificado em `-L`.

Verificou-se que a política de alocação do novo espaço é a mesma inicial, ou seja, `/dev/vgt1/lvs` expandiu no modo `stripe` e `/dev/vgt1/lv1` por concatenação. Verificou-se também que o LV `/dev/vgt1/lvs` não permitiu fazer uma nova expansão depois desta primeira, pois a partição `/dev/sdb` estava toda

ocupada. O volume com alocação por concatenação não teve este problema.

Após ser verificado o acréscimo de 1 GB, foram executados os seguintes comandos para diminuir os LVs e liberar espaço:

```
#umount /dev/vgt1/lvs
#e2fsadm -L-1G /dev/vgt1/lvs
#mount /dev/vgt1/lvs
#umount /dev/vgt1/lvl
#e2fsadm -L-1G /dev/vgt1/lvl
#mount /dev/vgt1/lvl
```

Os LVs `/dev/vgt2/lvlr` e `/dev/vgt2/lvsr`, que foram formatados com o sistema de arquivos reiserfs, foram alterados usando os seguintes comandos:

```
#lvextend -L-200M /dev/vgt2/lvsr
#resize_reiserfs -s -200M /dev/vgt2/lvsr
#lvextend -L+1G /dev/vgt2/lvlr
#resize_reiserfs -s +1G /dev/vgt2/lvlr
```

O comando `lvextend` estende o tamanho do volume lógico. O parâmetro `-L` redimensiona o sistema de arquivos para o tamanho especificado. Se for especificado `+` ou `-`, o tamanho é em relação ao espaço alocado, senão o volume fica do tamanho especificado em `-L`.

Conforme previsto, estas alterações foram feitas sem a necessidade de tornar os volumes indisponíveis.

4.7 - Adição de um novo disco ao VG

Para este exemplo, foi criada uma única partição no disco `/dev/sde`, de 14 GB. Após inicializar a partição com `pvcreate`, ela foi adicionada ao VG `vgt1`:

```
#vgextend vgt1 /dev/sde1
```

Verificou-se, que o novo tamanho de vgt1 passou a ser 20,88 GB, como pode ser observado no resultado do comando vgdisplay abaixo:

```
[root@sbcdf228 root]# vgdisplay vgt1
--- Volume group ---
VG Name          vgt1
VG Access        read/write
VG Status        available/resizable
VG #             2
MAX LV           256
Cur LV          2
Open LV          2
MAX LV Size      2 TB
Max PV           256
Cur PV          4
Act PV           4
VG Size          20.88 GB
PE Size          64 MB
Total PE         334
Alloc PE / Size  68 / 4.25 GB
Free PE / Size   266 / 16.62 GB
VG UUID          SfvLxg-gvqD-0PJA-p624-8d0N-zTLu-b6a9hz
```

4.8 – Movimentação de PVS

A movimentação de PVs às vezes é necessária para que um disco seja liberado ou quando um novo disco é acrescentado e alguns dados migrados para este disco. Outra finalidade da movimentação de PVs é a alteração da disposição dos dados nos discos de forma a melhorar o tempo de resposta. No exemplo, para migrar um dos VGs para outro servidor, como os dois VGs criados usavam os mesmos discos físicos, foi necessário mover as partições de forma que a retirada dos discos de vgt2 não afetasse a disponibilidade de vgt1. Para isto, foram feitos os seguintes procedimentos para mover os dados de /dev/sdb1 para /dev/sde1 e mover o volume /dev/sdb1 para vgt2:

```
[root@teste1 root]#pvmove /dev/sdb1 /dev/sde1
[root@teste1 root]# vgreduce vgt1 /dev/sdb1
[root@teste1 root]# vgextend vgt2 /dev/sdb1
```

Os mesmos procedimentos foram feitos para mover os dados de /dev/sdd2 para /dev/sdb1 e de /dev/sdc1 para /dev/sdd2. Desta forma, vgt1 passou a usar apenas os discos /dev/sdd e /dev/sde. Os outros discos, /dev/sdb e /dev/sdc passaram a ser usados apenas por vgt2, podendo ser transferidos para outro servidor sem afetar vgt1. Todos estes procedimentos foram executados com os sistemas de arquivos montados e disponíveis para leitura e gravação.

4.9 – Movimentação um VG para outro servidor

Uma funcionalidade interessante do LVM é a facilidade de mover VGs de um servidor para outro, no caso de necessidade de um servidor com maior desempenho, por exemplo. Para tal, foram utilizados os comandos vgexport e vgimport. Para teste, foi usado o vgt2.

O primeiro passo foi desmontar todos os sistemas de arquivos e tornar o vgt2 inativo. Só depois, o VG foi “transportado”.

```
[root@teste1 root]# vgchange -a n vgt2
vgchange -- volume group "vgt2" successfully deactivated
[root@teste1 root]# vgexport vgt2
vgexport -- volume group "vgt2" successfully exported
```

Após este comando, o VG passou a ser visto como “*exported volume*”.

Os discos foram retirados do servidor “teste1”, deixando de aparecer no sistema após o reboot.

Os discos foram transferidos para o servidor “teste2”, que após o boot passou a listá-los da seguinte forma quando emitido o comando pvscan:

```
[root@teste2 root]# pvscan
pvscan -- reading all physical volumes (this may take a while...)
pvscan -- inactive PV "/dev/sddlmba1" is in EXPORTED VG "vgt2"
[1.91 GB / 1.57 GB free]
pvscan -- inactive PV "/dev/sddlmba2" is in EXPORTED VG "vgt2"
[1.91 GB / 588 MB free]
pvscan -- inactive PV "/dev/sddlmbb2" is in EXPORTED VG "vgt2"
[1.91 GB / 1.57 GB free]
```

Para que o VG passasse a ser usado em teste2 foi efetuado o seguinte comando:

```
[root@teste2 root]# vgimport vgt2
Foi necessário ativar o volume no novo sistema, usando o comando:
[root@teste2 root]# vgchange -ay vgt2
```

Observou-se que, embora no outro sistema (teste2) os discos passaram a ter outros endereços, as informações sobre o VG continuaram no disco, o que permitiu a definição do vgt2 no novo servidor.

4.10 – Cópia de Segurança da Configuração do LVM

Quando se cria um dispositivo lógico no LVM, assim como acontece com quase todos os dispositivos no Linux, é criada uma entrada no diretório /dev. Por exemplo, ao criar o VG vgt1 foi criado o diretório /dev/vgt1. Ao criar os LVs nesse VG, foram criadas entradas em /dev/mapper que indicam se o

dispositivo é orientado a bloco ou a caracter. No diretório /dev/vgt1 foram criados links para estas entradas, como pode ser visto nos comandos ls a seguir:

```
# ls -l /dev/mapper/vgt1-lv01
```

```
brw----- 1 root root 253, 0 Jan 22 22:47 /dev/mapper/vgt1-lv01
```

```
# ls -l /dev/vgt1
```

```
total 0
```

```
lrwxrwxrwx 1 root root 21 Jan 22 22:47 lv01 -> /dev/mapper/vgt1-lv01
```

Como a configuração do LVM é mantida no VGDA de cada PV, caso estas entradas sejam perdidas por algum motivo, é possível recuperá-las, usando o comando `vgscan`, usado como abaixo:

```
#vgscan --mknodes
```

Este comando procura, em todos os dispositivos de armazenamento, PVs e VGs. Com o parâmetro `-mknodes`, ele verifica o diretório /dev e recria todos os diretórios de VG e as entradas especiais dos LVs que forem necessárias, a partir das informações contidas nos VGDA. No LVM2, este comando roda automaticamente toda vez que o equipamento é inicializado, porém ele pode ser rodado manualmente caso seja necessário.

Uma vez que toda a configuração do LVM é mantida nos PVs, é recomendável manter uma cópia destas configurações (metadados), para que seja possível recuperar os dados dos discos, caso alguma área de VGDA seja corrompida. Para isto, é usado o comando `vgcfgbackup`. Se o comando for executado sem parâmetros é feita uma cópia da configuração de todos os VGs no diretório /etc/lvm/backup. Com o parâmetro `-f` pode ser definido um nome para o arquivo de backup. Para fazer a cópia de configuração de um volume apenas, basta defini-lo no comando, como no exemplo a seguir, em que é feito o backup dos metadados de vgt1 no arquivo /bkp/vgt1:

```
# vgcfgbackup -f /bkp/vgt1 vgt1
```

Caso seja necessário recuperar os metadados do LVM, basta executar o comando `vgcfgrestore`, que usa os mesmos parâmetros do comando

vgcfbackup.

5 - CONCLUSÃO

O propósito do desenvolvimento deste trabalho foi apresentar os conceitos teóricos e um exemplo de implementação do LVM, mostrando as vantagens e desvantagens em relação a outras formas de particionamento de disco.

Os benefícios que o uso do LVM pode fornecer são extremamente importantes para um administrador de servidores Linux. A flexibilidade na administração do espaço em discos disponibilizada pelo LVM permite que um bom administrador aproveite melhor os dispositivos de armazenamento e obtenha melhor desempenho no acesso aos discos. Além disto, o uso do LVM torna os equipamentos mais disponíveis, por não exigir reinicializações para alterações em partições. Porém, para que estas facilidades sejam aproveitadas em sua totalidade, é necessário que o administrador conheça bem o funcionamento do LVM e o perfil de utilização de suas partições, de modo a alocar os espaços de forma a prover mais desempenho.

Para um melhor aproveitamento da facilidade de redimensionamento de partições do LVM, é necessário utilizar um sistema de arquivos que permita o redimensionamento online, como o reiserfs.

As ferramentas para movimentação de partições e a possibilidade de criação de LVs da forma *striped* permitem obter melhor desempenho no acesso aos discos através da melhor distribuição dos dados. Para isto, é preciso um acompanhamento constante do crescimento de utilização dos discos, em termos de quantidade de operações.

Como o maior ganho que se tem com o uso de LVM é a facilidade de administração do espaço em disco, em ambientes pequenos e estáveis, em que são utilizados poucos discos e o crescimento de cada partição é fácil de ser determinado com pequena possibilidade de erro, o uso de LVM não é recomendável.

Como pode ser visto no Capítulo 4, em que foi dado um exemplo de implementação, são poucos os comandos necessários para usar o LVM e muito simples. Desde que todos os conceitos relativos ao LVM estejam bem entendidos a implementação é fácil e rápida.

Como trabalhos futuros, pretende-se testar o uso de ferramentas gráficas para administração do LVM, com o objetivo de verificar o quanto o uso destas ferramentas pode auxiliar na administração dos volumes. Além disso, pretende-se verificar quantitativamente os efeitos do uso de LVM no desempenho, através da comparação entre o tempo de acesso com o uso de LVM e o tempo de acesso sem o uso.

Referências Bibliográficas

- [1] CLAUSEN, Andrew, KREUTER, M. The GNU Parted Manual. GNU Parted, v. 1.6.1, Maio 2002. [on-line]. Disponível na internet via www. url:
http://www.gnu.org/software/parted/manual/html_mono/parted.html#SEC1. Arquivo capturado em 01/09/2004.
- [2] [DD] DISKDRAKE. MandrakeSoft. [on-line]. Disponível na internet via www. Url: <http://www.mandrakelinux.com/diskdrake>. Arquivo capturado em 01/09/2004.
- [3] FERREIRA, Rubem E. Linux, Guia do Administrador do Sistema. Novatec Editora, 2003.
- [4] HARRIS, Tony, KOEHNTOPP, K. Linux Partition HOWTO, versão 2.4, novembro de 1997. Disponível na internet via www. Url: <http://www.lissot.net/partition/Partition.html>. Arquivo capturado em 19/08/2004.
- [5] HASENSTEIN, Michael. The Logical Volume Manager (LVM). SuSE Inc. 2001. [on-line]. Disponível na internet via www. Url: <http://www.novell.com/products/linuxenterpriseserver8/whitepapers/LVM.pdf>.
Arquivo capturado em 01/09/2004.
- [6] HINNER, Martin. Filesystems HowTO. Versão 0.7.5, agosto/2000. [on-line]. Disponível na internet via www. Url: <http://howtos.linux.com/howtos/Filesystems-HOWTO.shtml#toc6>.
Arquivo capturado em 21/12/2004.
- [7] LEWIS, AJ. LVM HOWTO, versão 0.14, 2004. Red Hat. [on-line]. Disponível na internet via www. Url: <http://www.ibiblio.org/pub/Linux/docs/HOWTO/other-formats/pdf/LVM-HOWTO.pdf>. Arquivo capturado em 10/02/2005.

- [8] MAUELSHAGEN, Heinz. LVM: Enterprise computing with the Linux Logical Volume Manager. Linux Magazine, 2002, número 16. [on-line]. Disponível na internet via www. Url: <http://www.linux-magazine.com/issue/16/LogicalVolumeManager.pdf>. Arquivo capturado em 30/07/2004.
- [9] Microsoft Windows XP Professional Resource Kit Documentation. Microsoft Corporation, 2004. [on-line]. Disponível na internet via www. Url: http://www.microsoft.com/resources/documentation/Windows/XP/all/reskit/en-us/Default.asp?url=/resources/documentation/Windows/XP/all/reskit/en-us/prpt_pt2_thoq.asp. Arquivo capturado em 20/09/2004.
- [10] Norton(tm) PartitionMagic(r) 8.0 Quick Start Guide. Symantec Corporation, 2004. [on-line]. Disponível na internet via www. Url: ftp://ftp.symantec.com/public/english_us_canada/products/norton_partitionmagic/npmagic_8/manuals/norton_partitionmagic_8Quick.pdf. Arquivo capturado em 20/09/2004.
- [11] PartitionExpert 2003 User's Guide. Swsoft, 2002. [on-line]. Disponível na internet via www. url: http://www.acronis.com/pdf/partitionexpert2003_ug.en.pdf. Arquivo capturado em 20/09/2004.
- [12] Revista do Linux. [on-line]. Conectiva, abril de 2001. Número 16. Disponível na internet via www. url: <http://www.revistadolinux.com.br/ed/016/assinantes/particionamento.php3>. Arquivo capturado em 01/09/2004.
- [13] ROBBINS, Daniel. Common Threads: Learning Linux LVM, Part 1. IBM, março de 2002. [on-line]. Disponível na internet via www. Url: <http://www-106.ibm.com/developerworks/linux/library/l-lvm/#h0>. Arquivo capturado em 30/07/2004.

- [14] SILVA, Gleydson M. da. Fdisk-Portugues-HOWTO V. 1.36, 1999.
[on-line]. Disponível na internet via www. url:
<http://www.terravista.pt/ilhadomel/5172/fdisk-portuguese.htm#1.1>.
Arquivo capturado em 01/09/2004.
- [15] TEIXEIRA, Roberto & MERCER, Carlos Daniel. Guia do Servidor.
Conectiva, 2000. [on-line]. Disponível na internet via www. Url:
http://linorg.cirp.usp.br/Guias.Conectiva/Guias_V.6.0/guia_servidor/index.html. Arquivo capturado em 11/02/2005.
- [16] TORRES, Bruno. Particionamento de disco. Abril de 2004. [on-line]. Disponível na internet via www. Url:
<http://www.brunotorres.net/gnulinux/particionamento-disco>. Arquivo capturado em 27/05/2004.
- [17] VINAYAK, Hedge. Using the Logical Volume Manager. [on-line]. Disponível na internet via www. Url:
<http://ldp.rtin.bz/LDP/LGNET/issue84/vinayak.html>. Arquivo capturado em 05/03/2005.