

4.1 概要

論理ボリューム管理 (LVM) システムもディスク・アレイ (RAID) もストレージの信頼性を上げるものですが、論理的な信頼性を上げるものであって、ハードウェアの信頼性を上げるものではありません。したがって、ハードウェアで何か障害が発生した場合は、LVMやRAIDでその信頼性を補完している間に物理的な障害を取り除くことが急務です。

特に、コンピュータ・システムの最も基本的なインフラであるストレージの信頼性はシステム自体の信頼性に直結するので、管理者は最も敏感であるべきです。

なお、ディスクのローレベルの処理を取り扱うため、設定作業には十分な注意が必要です。もし、設定に間違いがあると、多くの場合、システム起動時にシステム動作不能になる、というような重大なエラーが発生し、ファイル・システムがRead-Onlyでマウントされているため、修復が不可能になる事態となります^(*4.1)。

ファイル・システム・テーブル (fstab) を設定変更することが多く、その場合、起動時にエラーとなると以降の処理ができないので、fstabを変更する前に試しに同じ設定で一時的にマウントしてみしてから変更するなどのテストが重要です。

こうしたことから、この章で行うLVMやRAIDの設定・変更などを含む処理は、すべてシングル・ユーザ・モードで行うこととなります。

4.2 論理ボリューム管理：LVM

LVM^(*4.2)では、複数の物理ディスク領域を横断的に利用した論理ボリュームにより、さまざまなI/Oの出し方が可能になります。複数の領域に順繰りにI/Oを出したり、一つの領域へのI/Oをほかの領域にミラーしたりするなどです。この前者が、I/Oの効率化と負荷分散を提供するもので、後者がミラーリングによる冗長化になります。

LVMファイル・システムは、Linuxインストール時に初期化設定して作成することも、インストール後(つまり、システム運用中)に作成したり、変更したりすることも、いずれも可能です。

^(*4.1) こうした場合、Linuxの「Live CD」やKNOPPIXなどのCDシステムや、あらかじめ作成したCDイメージなどから起動するか、あるいは、別システムにディスクを一時的に接続して起動して、修正を行うなど手の込んだ処理が必要となる。

^(*4.2) LVM : Logical Volume Manager

この章では、LVMの構造、Linuxインストール中のLVMの作成、システム運用中のLVMの作成の順に説明します。また、LVMの三つのタイプの作成方法なども解説します。

4.2.1 LVMの構造

LVMの主要な構成要素は、LV (Logical Volume；論理ボリューム)、VG (Volume Group；ボリューム・グループ)、PV (Physical Volume；物理ボリューム)、そして、PE (Physical Extent；物理エクステント)の四つです。Linuxファイル・システムはLVの上で作成されます。

図4.1のように、一般のディスク・ドライブはPVにマッピングされ、このPVを構成する最小単位がPEです。VGは1または複数のPVの集まりから成り、このVGの中で1または複数のLVが1または複数のPEを使用することになります。図のように、一つのLVは複数のPEを占有しますが、これらのPEが複数のPVに渡ることも可能です。

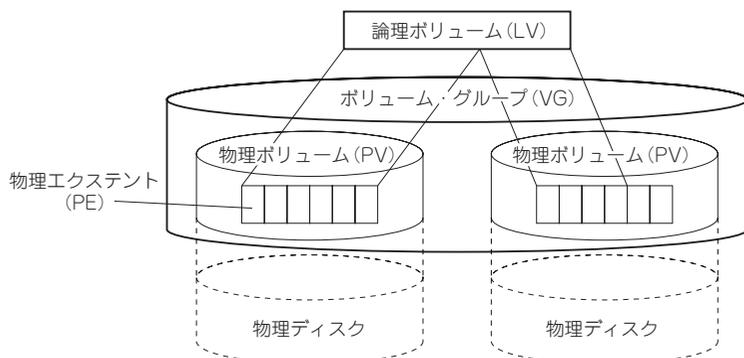


図4.1 LVMの構造

なお、PVの構成は、図4.2のように、通常の一つのLinuxパーティション内を四つの領域で構成されるように設定されます。パーティションタイプは「8e」(LVM)です。

第1セクタ
第2セクタ=LVMラベル
メタデータ領域
ユーザ・データ領域

図4.2 PVの構成

LVMの特徴は、その上に構築されたファイル・システムの拡張性にあります。ファイル・システムの容量が大きくなり、パーティションが手狭になったときには、一般にシンボリック・リンクで別パーティションにリンクさせて急場をしのぎますが、あまり

多くなるとシステム効率が低下します。一方、LVMではPVを追加(ボリューム増設)してLVを拡張することでシステム効率を低下させることがなく、拡張性に富みます。

なお、ファイル・システムの/bootは、LVM化できません^(*4.3)。

4.2.2 LVのタイプ

LVには図4.3のような三つのタイプがあります。リニアLV(Linear LV)(デフォルト)、ストライプLV(Striped LV)、そして、ミラーLV(Mirrored LV)です。

リニアLVは、PEを順に利用していくタイプですが、ストライプLVはデータ書き込みをPV上でストライプ単位に(一部ずつ層単位に)行います。ミラーLVでは一つのPVへの書き込みと全く同じコピーを別のPVにも行います(ミラーリング)。

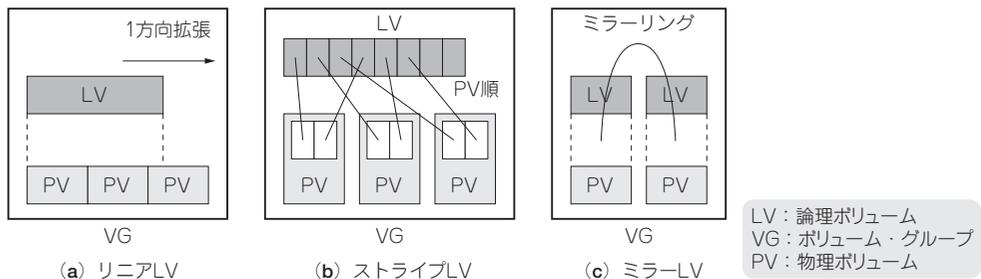


図4.3 論理ボリュームLVのタイプ

●リニアLV

リニアLVは、VG内のPVを複数使用することもあります。その場合、これらの複数のPVのそれぞれの利用可能サイズ(PE数)は同じである必要はありません。

リニアLVは、ほかのタイプのLVに比べて特異性はありませんが、次節で解説する、ミラーリング(RAID-1)と併用すれば、拡張性+冗長性を手に入れることができます。

●ストライプLV

ストライプLVでは、複数のPVに対してはデータ書き込みを順繰りに分割して行うため、異なるドライブ上の複数のPVに対して適用すると、ストライプLV上のファイル・システムではI/Oの並行処理によりI/Oの効率向上・負荷分散が望めます。当然のことですが、同じドライブ上で行うとその効果は望めません。

●ミラーLV

ミラーLVでは、異なるドライブ上のPVにミラーリングを適用することで、そのファ

(*4.3) Linuxインストール時のLVMパーティションでは、/bootがファイル・システムがなく、/bootをLVM化するとエラーになる。

イル・システムに対して冗長化，フォールト・トレランスの実現が可能になります。次節のミラーリング (RAID-1) にLVMを適用することで同じことが可能ですが，一つのLV内でミラーを意識することなく利用することができます。

ミラーLVでは最大一つのPVまでのサイズで利用することができます。一方，ミラーリング (RAID-1) にLVMを適用した場合，複数のミラー・ボリュームに渡ってLVが利用可能になります。

4.2.3 Linuxインストール時のLVMファイル・システムの作成

Linuxインストール時の初期化作成では，LVタイプはデフォルトでリニアLVとなります。

ここでは，詳細な解説はしませんが，インストール時のディスク・パーティションの設定のファイル・システム・タイプで「physical volume (LVM)」を選択してから，「LVM」ボタンで，実際に使用するファイル・システムを設定する，という2段階で行います。

実際のファイル・システムは，LVM設定したパーティション (PV) から利用するものを選択してVGを設定し，その中でLVを追加編集していきます。

図4.4と図4.5はLinuxインストール時に作成したLVMを論理ボリューム管理で見たものです。



図4.4 論理ボリューム管理で見たインストール後のボリューム・グループ00の構造