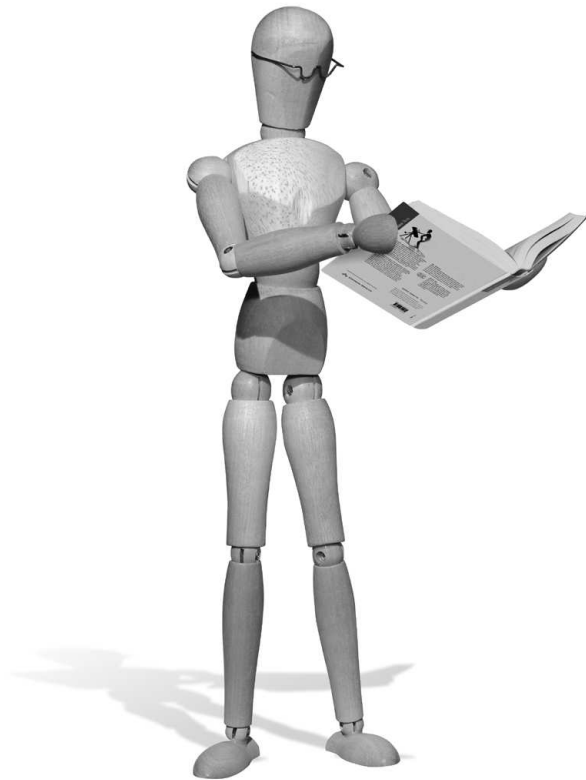


Teil I

Installation



Im ersten Teil dieses Buchs geht es um die Installation eines Servers auf der Basis der Ubuntu-Server-CD. Vielleicht denken Sie sich jetzt, zum Thema Linux-Installation sei in unzähligen Büchern und Artikeln in den letzten 20 Jahren schon genug geschrieben worden. Da gebe ich Ihnen grundsätzlich natürlich recht – aber die Server-Installation hat dennoch ihre Eigenheiten.

Wenn Ihnen nicht von vorneherein klar ist, wie RAID und LVM funktionieren und welche Vor- und Nachteile damit verbunden sind, werden Sie diese wahrscheinlich erst nach einigen Monaten Server-Betrieb erkennen. Dann ist es aber zu spät: Ein nachträgliches Einrichten von RAID oder LVM ist zumeist unmöglich oder zumindest mit großem Aufwand verbunden. Deswegen beginnt das Buch mit einem Grundlagenkapitel. Es erklärt Partitionstypen, RAID (*Redundant Array of Inexpensive/Independent Disks*) und LVM (*Logical Volume Manager*).

Das zweite Kapitel beschreibt die Ubuntu-Server-Installation mit RAID und LVM und hilft Ihnen, sich in den verschachtelten Dialogen des Installationsprogramm, nicht zu verirren.

Wer bisher unter Linux primär mit einer grafischen Benutzeroberfläche gearbeitet hat, findet in Kapitel 3 einige Tipps, um sich in der Textkonsole oder einem SSH-Terminal zurechtzufinden.



open source library

1 RAID- und LVM-Grundlagen

Normalerweise bemühe ich mich, ein Buch so »locker« wie möglich zu beginnen – z. B. mit einem *Hello world!*-Beispiel. Diesmal muss ich aber eine Ausnahme machen: Bevor ich im nächsten Kapitel die Details einer Server-Installation erkläre, gehe ich hier auf die erforderlichen Grundlagen ein.

Leider sind Festplattenpartitionierung, RAID, LVM & Co. keine einfachen Themen. Daher rate ich Ihnen, dieses Kapitel in Ruhe zu studieren: Sämtliche während der Installation getroffenen Entscheidungen, die die Nutzung der Festplatte(n) betreffen, sind nur mit großem Aufwand rückgängig zu machen – oft nur durch eine Neuinstallation. Anders formuliert: Dieses Kapitel kostet Sie jetzt vielleicht ein, zwei Stunden Zeit. Es erspart Ihnen aber unter Umständen mehrere Tage Arbeitsaufwand, der sonst vielleicht nach ein paar Wochen oder Monaten bei der Neukonfiguration Ihres Servers erforderlich wäre ...

Dieses Kapitel geht nur auf dateisystemspezifische Grundlagen ein, wobei der Schwerpunkt bei RAID und LVM liegt. Installationsdetails folgen im nächsten Kapitel. Kommandos zur Administration des Dateisystems eines laufenden Systems stelle ich in Kapitel 8 vor.

Ich gehe in diesem Buch davon aus, dass Sie auf Ihrem Rechner ausschließlich Ubuntu installieren. Parallelinstallationen mit anderen Linux- und Windows-Betriebssystemen sind bei einem Server nicht zweckmäßig: Es würde die Funktion eines Servers ad absurdum führen, wenn Sie den Rechner zwischendurch unter Windows starten: Alle anderen Rechner im Netzwerk hätten plötzlich keinen Serverzugang mehr.

1.1 Physikalische Partitionierung der Festplatte

Eine Festplatte kann in mehrere Abschnitte (Partitionen) gegliedert werden. Das ist notwendig, wenn Sie mehrere Betriebssysteme parallel installieren oder wenn Sie verschiedene Datenbereiche voneinander trennen möchten. Für eine Linux-Server-Installation brauchen Sie mindestens zwei Partitionen: eine kleine für den Swap-Bereich und eine große für das Systemverzeichnis (root-Verzeichnis). Es kann auch

Gründe geben, das Systemverzeichnis über mehrere Partitionen zu verteilen. Details dazu folgen im nächsten Abschnitt ab Seite 19. Vorerst geht es aber darum, welche Regeln bei der Partitionierung der Festplatte einzuhalten sind.



Achtung

Nachträgliche Änderungen an der Partitionierung einer Festplatte sind nur sehr schwer möglich. Jede Größenänderung hat normalerweise den Verlust aller Daten auf dieser Partition zur Folge. Es gibt zwar Situationen, in denen sich das umgehen lässt (indem nicht nur die Größe der Partition, sondern auch die des darin enthaltenen Dateisystems verändert wird), aber in jedem Fall ist eine nachträgliche Änderung der Grundstruktur aufwendig, fehleranfällig und ganz einfach lästig. Um derartige Änderungen von vornherein zu vermeiden, sollten Sie sich einerseits schon vor der Installation Gedanken darüber machen, wie Ihr Server in den nächsten zwei, drei Jahren wachsen soll, und andererseits den Einsatz von LVM in Erwägung ziehen (siehe Seite 28).

Partitionstypen

Es gibt drei Typen von Festplattenpartitionen: primäre, erweiterte und logische Partitionen. Auf einer Festplatte können maximal vier primäre Partitionen existieren.

Statt einer primären Partition können Sie auch eine erweiterte Partition anlegen. Die erweiterte Partition kann zwar selbst keine Daten enthalten, dafür aber mehrere logische Partitionen. Logische Partitionen können dann wie primäre Partitionen verwendet werden. Der Sinn von erweiterten und logischen Partitionen besteht einzig darin, das historisch vorgegebene Limit von nur vier primären Partitionen zu umgehen.

Die Anzahl der logischen Partitionen ist je nach Betriebssystem limitiert. Linux kann in der aktuellen Version auf maximal elf logische Partitionen zugreifen. Bei optimaler Einteilung der Festplatte ergibt sich somit ein Limit von 14 für Daten verwendbare Partitionen pro Festplatte (drei primäre, eine erweiterte und elf logische Partitionen). In der Praxis stellt das nur selten eine Einschränkung dar – und wenn doch, können Sie das Limit mit LVM umgehen (siehe Abschnitt 1.5).

Partitionsnamen

Unter Linux erfolgt der interne Zugriff auf Festplatten bzw. deren Partitionen über sogenannte Device-Dateien: Die Festplatten erhalten der Reihe nach die Bezeichnung `/dev/sda`, `/dev/sdb`, `/dev/sdc` etc.

Um eine einzelne Partition und nicht die ganze Festplatte anzusprechen, wird der Name um die Partitionsnummer ergänzt. Die Zahlen 1 bis 4 sind für primäre und erweiterte Partitionen reserviert. Logische Partitionen beginnen mit der Nummer 5

1.2 Partitionierung für den Server-Betrieb

– auch dann, wenn es weniger als vier primäre oder erweiterte Partitionen gibt. Abbildung 1.1 veranschaulicht die Nummerierung: Auf der Festplatte gibt es zwei primäre Partition und eine erweiterte Partition, die drei logische Partitionen enthält.

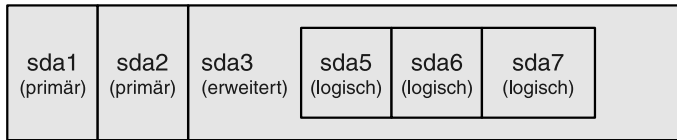


Abbildung 1.1: Linux-Partitionsnamen

In der Praxis werden Sie normalerweise die ersten drei Partitionen als primäre Partitionen anlegen. Brauchen Sie dann noch mehr Partitionen, erzeugen Sie eine erweiterte Partition und darin so viele logische Partitionen wie erforderlich. Manche Partitionierungshilfen nehmen Ihnen etwas Denkarbeit ab und unterscheiden nicht zwischen primären und logischen Partitionen. Stattdessen legen sie erweiterte Partition bei Bedarf selbstständig an.

1.2 Partitionierung für den Server-Betrieb

Sie wissen nun, dass Sie eine Festplatte in viele Partitionen zerlegen können. Aber warum brauchen Sie überhaupt mehrere Partitionen für eine Server-Installation, bei der ja nur eine einzige Linux-Distribution auf dem Rechner installiert wird? Die folgende Aufzählung beschreibt der Reihe nach den Zweck verschiedener Partitionen, wobei Sie aber durchaus nicht immer alle hier genannten Partitionen wirklich benötigen.

Systempartition

Die Systempartition ist die einzige Partition, die unbedingt erforderlich ist. Sie nimmt das Linux-System mit all seinen Programmen und Daten auf. Diese Partition wird immer das Verzeichnis / zugeordnet. Dieses Verzeichnis ist der Punkt, an dem die Partition in das Dateisystem eingebunden wird (der sogenannte mount-Punkt). Wenn der Server einmal läuft, sprechen Sie diese Partition mit dem Pfad / an. / bezeichnet die Wurzel, also den Anfang des Dateisystems. Aus diesem Grund wird die Systempartition oft auch als Root-Partition bezeichnet.

Swap-Partition

Die Swap-Partition ist das Gegenstück zur Auslagerungsdatei von Windows: Wenn Linux zu wenig RAM hat, lagert es Teile des gerade nicht benötigten RAM-Inhalts dorthin aus. Die Verwendung einer eigenen Partition (statt wie unter Windows einer gewöhnlichen Datei) hat vor allem Geschwindigkeitsvorteile. Linux kann zwar statt einer Swap-Partition auch eine Swap-Datei verwenden, das ist aber unüblich und langsam.

Darüber, ob eine Swap-Partition bei Rechnern mit viel RAM überhaupt sinnvoll ist, hat es schon lange Diskussionen gegeben. Grundsätzlich funktioniert Linux auch ohne Swap-Speicher; ein Argument spricht aber für eine Swap-Partition: Sollte eines Ihrer Programme außer Kontrolle geraten oder aus anderen Gründen mehr Speicher brauchen als erwartet, ist der verfügbare Speicher irgendwann erschöpft. Das führt zum Absturz des nächsten Prozesses, der weiteren Speicher anfordert. Das kann irgendein Prozess sein, nicht unbedingt Ihr außer Kontrolle geratenes Programm.

Grundsätzlich ändert eine Swap-Partition nichts an diesem Problem – auch der Swap-Speicher ist ja begrenzt. Durch die immer intensivere Nutzung des Swap-Speichers laufen alle Programme aber immer langsamer. Diese »automatische Notbremse« gibt oft die Möglichkeit, einzugreifen, bevor es zu spät ist. Persönlich richte ich für alle Linux-Systeme einen Swap-Bereich ein – egal ob der Rechner für den Desktop- oder Server-Einsatz konzipiert ist. Wenn Sie sich intensiver mit dem Thema befassen möchten, hier zwei Leseempfehlungen:

<http://www.thomashertweck.de/linuxram.html>
<http://kerneltrap.org/node/3202>

Weitere Datenpartitionen

Auf die gleiche Weise, wie Sie das Verzeichnis `/boot` in eine eigene Partition verlagern und aus der Systempartition herauslösen können, können Sie auch für andere Verzeichnisse verfahren. Typische Kandidaten sind in Tabelle 1.1 zusammengestellt.

<code>/home</code>	benutzerspezifische Daten (Samba, NFS)
<code>/tmp</code>	Austausch temporärer Dateien; zumeist geringer Platzbedarf
<code>/var</code>	veränderliche Daten
<code>/var/cache</code>	Zwischen- bzw. Pufferspeicher (alle Programme)
<code>/var/lib/mysql</code>	Datenbankdateien (MySQL)
<code>/var/log</code>	Logging-Dateien (alle Programme)
<code>/var/www</code>	Webdateien (Apache)

Tabelle 1.1: Wichtige Verzeichnisse für veränderliche Daten

In welchen Verzeichnissen der Platzbedarf am schnellsten wächst, ist ausschließlich von der individuellen Nutzung des Servers abhängig. Bei einem Dateiserver für das Netzwerk ist es `/home`, bei einem Webserver `/var/www`, bei einem Datenbankserver `/var/lib/mysql` etc.



Hinweis

Linux-Einsteiger, aber auch überraschend viele fortgeschrittene Linux-Benutzer kennen nicht den Unterschied zwischen den Verzeichnissen `/usr` und `/var`. Dabei ist es ganz einfach: `/usr` enthält die Programmdateien und -bibliotheken. `/var` ist für alle veränderlichen Daten reserviert. Mit anderen Worten: Nachdem die Installationsarbeiten abgeschlossen sind, wächst der Platzbedarf für das `/usr`-Verzeichnis nicht mehr. Ganz anders sieht es beim `/var`-Verzeichnis aus: Sein Platzbedarf wächst stetig.

In der Vergangenheit gab es die Idee, für das `/usr`-Verzeichnis eine eigene Partition vorzusehen und diese nach Abschluss der Installation in einem *Read-only*-Modus zu nutzen. Da diese Vorgehensweise Updates unmöglich macht, ist man von dieser Idee wieder abgekommen. Mein Tipp: Belassen Sie das `/usr`-Verzeichnis in der Systempartition!

Zweck der Partitionierung

Bevor Sie eifrig unzählige Datenpartitionen anlegen, sollten Sie sich fragen: Wozu? Selbst wenn Sie LVM einsetzen (siehe Seite 28), führt die Aufteilung des Systemverzeichnisses über mehrere Datenpartitionen zu einem erheblichen zusätzlichen Administrationsaufwand.

Das gebräuchlichste Argument für die Zerlegung ist die Möglichkeit der Geschwindigkeitsoptimierung: So können Sie für eine Partition, die überwiegend kleine Dateien enthält, einen speziell dafür optimierten Dateisystemtyp verwenden (z. B. `reiserfs`). Auch die Verteilung der Partitionen über mehrere Festplatten kann zweckmäßig sein (etwa bei einem Datenbankserver zur physikalischen Trennung von Daten- und Loggingdateien). In der Praxis fällt der Geschwindigkeitsgewinn aber meist bescheiden aus.

Ein weiteres Argument ist der bereits erwähnte Zeitaufwand für die Überprüfung großer Dateisysteme. Indem Sie statt einer riesigen Systempartition mehrere kleinere Partitionen verwenden, ist der Zeitaufwand pro Partition wesentlich geringer. Wenn Sie alle Partitionen gleichzeitig überprüfen müssen, haben Sie natürlich nichts gewonnen. Aber es besteht nun die Möglichkeit, einzelne Datenpartitionen im laufenden Betrieb zu überprüfen und während dieser Zeit nur einzelne Server-Dienste abzuschalten (anstatt den ganzen Server offline zu stellen).

Bei Desktop-Installationen empfehle ich immer, das `/home`-Verzeichnis in einer eigenen Partition anzulegen. Damit können Sie später eine neue Ubuntu-Version oder auch eine ganz andere Distribution installieren, ohne Ihre Daten anzutasten. Bei einer Server-Installation gilt dieser Ratschlag allerdings nur mit Einschränkungen: Hier gilt zumeist die Regel *Never change a running system!* Eine Neuinstallation wird daher oft nur durchgeführt, wenn es wirklich zwingende Gründe dafür gibt (etwa weil es für die installierte Distribution keine Updates mehr gibt). Selbst dann

wird man wohl zuerst ein Distributions-Update versuchen, um eine komplette Neu-konfiguration zu vermeiden (also z. B. von Ubuntu 8.04 LTS auf Ubuntu 10.04 LTS). In vielen Fällen ist eine Neuinstallation oder ein grundlegendes Update ohnedies ein guter Zeitpunkt, auch die Hardware zu ersetzen. Das gibt Ihnen die Möglichkeit, das alte System so lange weiterlaufen zu lassen, bis das neue System komplett aufgesetzt ist und störungsfrei läuft. Das Server-Update bzw. die Neuinstallation lässt sich so nahezu unterbrechungsfrei durchführen.

Partitionsgrößen

Wie groß sollen die einzelnen Partitionen sein? Hierfür gibt es keine allgemeingültigen Antworten – zu sehr hängt die Antwort davon ab, wie Sie Ihren Server einsetzen. Dieser Abschnitt fasst dennoch einige Tipps zusammen.

- **Swap-Partition:** In der Vergangenheit lautete die Empfehlung, etwa das Zweifache des RAMs als Swap-Speicher vorzusehen. Mit zunehmender RAM-Größe ist diese Faustregel aber selten zielführend: Oft reicht die einfache RAM-Größe. Beachten Sie aber, dass Sie Ihren Server vielleicht in ein, zwei Jahren mit mehr RAM ausstatten werden. Die Swap-Partition können Sie dann nur schwer vergrößern. Insofern ist eine große Swap-Partition dann von Vorteil.

Sehr konkrete Empfehlungen gibt Oracle für den Einsatz seines Datenbank-servers. Je nach verfügbarem RAM kommen unterschiedliche Faktoren zur Berechnung der optimalen Swap-Größe zur Anwendung:

Bis 2 GByte RAM:	Faktor 2
2 bis 8 GByte RAM:	Faktor 1
Mehr als 8 GByte RAM:	Faktor 0,75

Wesentlich großzügiger ist der Ubuntu-Installer, wenn er die Swap-Partition selbst einrichtet: Auf einem Testrechner mit 2 GByte RAM erreicht die Swap-Partition dann stattliche 6 GByte.

- **Bootpartition:** Die Bootpartition enthält anfänglich nur den Kernel, eine sogenannte Initrd-Datei sowie ein paar Konfigurationsdateien (ca. 20 bis 30 MByte). Mit jedem Kernel-Update kommt ein neues Set von Kernel- und Initrd-Dateien hinzu (ca. weitere 10 bis 15 MByte). Einige 100 MByte reichen für diese Partition also vollkommen aus.

Wenn Sie ganz auf Nummer sicher gehen möchten, reservieren Sie ein GByte – das reicht für ein monatliches Update über fünf Jahre. (Natürlich können Sie auch hin und wieder einen Blick in das /boot-Verzeichnis werfen und alte Kernel-Dateien löschen. Aber erfahrungsgemäß werden Sie daran erst denken, wenn es schon zu spät ist ...)

- **System- und Datenpartitionen:** Die Systempartition sowie alle eventuell davon abgespaltenen Daten- und Backup-Partitionen füllen den verbleibenden Rest der Festplatte(n). Wenn Sie sich für eine Aufteilung in mehrere Datenpartitionen entschieden haben, empfiehlt es sich, LVM einzusetzen und einen Teil der Festplatte(n) für spätere Vergrößerungen einzelner Partitionen zurückzuhalten.

1.3 Linux-Dateisystemtypen

Bevor Sie Daten in einer Partition speichern können, müssen Sie dort ein Dateisystem einrichten. Die einzige Ausnahme ist die Swap-Partition, die direkt ohne den Umweg über ein Dateisystem benutzt wird.

Unter Linux stehen verschiedene Dateisystemtypen zur Auswahl: ext2, ext3, jfs, reiserfs, xfs und in naher Zukunft auch noch ext4 und btrfs. Jeder dieser Dateisystemtypen hat Vor- und Nachteile, auf die ich hier aber nicht eingehen werde. Stattdessen empfehle ich Ihnen momentan, für Datenpartitionen den ext3-Dateisystemtyp einzusetzen.

ext3 ist nicht umsonst der bei Weitem populärste Linux-Dateisystemtyp: Er besticht durch eine sehr hohe Stabilität und Datensicherheit, hohe Performance in den meisten Anwendungsfällen und durch Journaling-Funktionen, dank derer das Dateisystem nach einem Stromausfall oder Totalabsturz sehr rasch wieder in einen konsistenten Zustand gebracht werden kann. (Anders als vielfach behauptet wird, stellen die Journaling-Funktionen aber keinen hundertprozentigen Schutz gegen Datenverluste dar!)

Die hohe Verbreitung von ext3 ist zudem ein Garant dafür, dass der Code weitgehend frei von Fehlern ist und dass eventuell doch noch vorhandene Fehler sehr rasch korrigiert werden. Gerade das Argument der langfristigen Wartung spricht zurzeit gegen die meisten anderen Dateisysteme – so bestechend deren technische Eigenschaften auch sein mögen. Schließlich steht mit ext4 bereits ein abwärtskompatibler Nachfolger vor der Tür, der noch größere Dateisysteme und in bestimmten Fällen eine höhere Geschwindigkeit verspricht.

Fundamentale Mängel

Alle hier aufgezählten Dateisysteme leiden unter zwei fundamentalen Problemen: Erstens sind sie nicht für den Einsatz von Flash-Speicher optimiert (also für Solid-State-Disks, SSDs). Diese neue Speichertechnologie wird vermutlich in naher Zukunft auch im Server-Bereich Einzug halten. SSDs haben ganz andere physikalische Eigenschaften als herkömmliche Festplatten und können daher mit speziellen Treibern bzw. Dateisystemtypen optimal genutzt werden. Ein kleiner Trost mag sein, dass auch andere Betriebssystementwickler noch keine konkreten Lösungen für dieses Problem anbieten.

Zweitens erfordert eine Dateisystemüberprüfung bei großen Datenpartitionen Stunden. Die Überprüfung kann nur stattfinden, während der Rechner gerade gestartet wird, nicht aber im laufenden Betrieb. Alle oben aufgezählten Dateisysteme mit der Ausnahme von ext2 haben Journaling-Funktionen, die eine vollständige Dateisystemüberprüfung nur selten erforderlich machen. Wenn das aber doch einmal der Fall ist oder wenn das Dateisystem routinemäßig überprüft wird (je nach Konfiguration nach einer bestimmten Zeit bzw. nach einer bestimmten Anzahl von Systemstarts), ist Ihr Server stundenlang offline. Immerhin arbeiten die Dateisystem-

entwickler bereits an diesem Problem: ext4 verspricht laut Entwicklerangaben eine Verbesserung um den Faktor sechs. Ein Hoffnungsschimmer ist zudem die Online-Dateisystemüberprüfung des btrfs-Dateisystems. Leider sind beide Dateisysteme momentan noch nicht praxistauglich.

1.4 RAID

RAID steht für *Redundant Array of Inexpensive/Independent Disks* (es gibt zwei Definitionen). Die Grundidee besteht darin, Partitionen mehrerer Festplatten logisch miteinander zu verknüpfen. Das Ziel ist dabei, ein zuverlässigeres und/oder schnelleres Gesamtsystem zu schaffen:

- Durch RAID kann die Datenübertragung gesteigert werden, da der Datenzugriff quasi parallel von mehreren Festplatten erfolgt.
- Durch RAID kann aber auch die Sicherheit gesteigert werden, da Daten redundant (mehrfach) gespeichert werden. Das ist dann allerdings mit Geschwindigkeitseinbußen verbunden und beansprucht zusätzlichen Speicherplatz.

Dieser Abschnitt beschreibt nur die RAID-Grundlagen. Details zur Installation und späteren Administration eines LVM-Systems folgen in den Kapiteln 2 und 8. Weitere Informationen finden Sie hier:

<http://de.wikipedia.org/wiki/RAID>

RAID-Level

Im folgenden Überblick über die wichtigsten RAID-Level gehe ich davon aus, dass jeweils gleich große Partitionen *unterschiedlicher* Festplatten miteinander verbunden werden. Theoretisch können auch Partitionen *einer* Festplatte verbunden werden; das ist aber weder vom Sicherheits- noch vom Geschwindigkeitsstandpunkt her sinnvoll. Neben den hier genannten RAID-Leveln gibt es noch einige weitere Varianten, die in der Praxis aber seltener vorkommen.

- **RAID-0 (Striping):** Bei RAID-0 werden mehrere physikalische Partitionen zu einer größeren Partition vereint. Dabei werden die Daten quasi parallel in kleinen Blöcken (z. B. 4 kByte) auf die einzelnen Partitionen verteilt, sodass die Daten beim Zugriff alternierend von allen Festplatten gelesen werden. Daraus ergibt sich im optimalen Fall eine Vervielfachung der Datenrate (d. h. bei drei Festplatten eine Verdreifachung). In der Praxis ist der Effekt meist kleiner als erhofft und kommt nur bei großen Dateien wirklich zum Tragen. Die Anzahl der Random-Access-Zugriffe pro Sekunde wird durch das Striping nicht verbessert.

RAID-0 hat einen gravierenden Nachteil: Das Ausfallrisiko ist hoch, weil *eine* defekte Festplatte zum Verlust *aller* Daten führt.

- **RAID-1 (Mirroring):** Bei RAID-1 werden dieselben Daten in der Regel auf zwei Partitionen gespeichert (selten auf mehr Partitionen). Wenn eine Festplatte ausfällt, stehen alle Daten auf den anderen Festplatten zur Verfügung. Der Vorteil ist die höhere Sicherheit, der Nachteil die halbierte Kapazität. Die Geschwindigkeit hängt von der Implementierung ab: Bei einem Hardware-RAID kann bei Schreiboperationen weitgehend die gleiche Geschwindigkeit wie bei einem normalen Zugriff auf eine Partition erreicht werden, und bei Lese- und Suchoperationen ist sogar eine Beschleunigung möglich. Bei einem Software-RAID sind aber insbesondere Schreibvorgänge etwas langsamer. (Die Unterschiede zwischen Hardware- und Software-RAID werden etwas weiter unten beschrieben.)
- **RAID-10:** RAID-10 kombiniert RAID-1 und RAID-0 und setzt mindestens vier Festplatten bzw. Partitionen voraus: Partition 1 und 2 bilden einen RAID-1-Verbund, Partition 3 und 4 einen weiteren RAID-1-Verbund. Auf der nächsten Ebene werden die zwei RAID-1-Partitionen nun zu einem RAID-0-Verbund kombiniert. Damit kombiniert RAID-10 die Vorteile von RAID-0 (Geschwindigkeit) und RAID-1 (Sicherheit).
- **RAID-5 (Parity Striping):** RAID-5 funktioniert im Prinzip wie RAID-0, allerdings werden zusätzlich in einer (für jeden Datenblock wechselnden) Partition Paritätsinformationen gespeichert. Wenn eine Festplatte ausfällt, können die gesamten Daten rekonstruiert werden. Das setzt zumindest drei Festplatten voraus. (Den Ausfall von zwei oder mehr Festplatten führt allerdings zu einem kompletten Datenverlust.)

RAID-5 ist ebenso sicher wie RAID-1 und bei Lesezugriffen etwa so schnell wie RAID-0. Zudem hat RAID-5 den Vorteil, dass die für die Redundanz erforderliche Datenanteil mit der Anzahl der Festplatten kleiner wird: Bei RAID-1 beträgt der Kapazitätsverlust immer 50 Prozent, bei RAID-5 beträgt er nur 33 Prozent bei drei Festplatten, 25 Prozent bei vier, 20 Prozent bei fünf etc.

RAID-5 hat gegenüber RAID-1 allerdings auch Nachteile: Zum einen sind Schreiboperationen langsamer als bei RAID-1, insbesondere wenn sich häufig kleine Datenmengen ändern. Der Grund ist, dass selbst bei kleinen Veränderungen die Paritätsinformationen für einen ganzen Datenblock neu berechnet und gespeichert werden müssen.

Nach dem Austausch einer defekten Platte dauert die Rekonstruktion des RAID-5-Verbunds sehr lange (viel länger als bei RAID-1). Bei einem Software-RAID kann der RAID-Verbund während dieser Zeit nicht genutzt werden. Sollte während der Rekonstruktion eine weitere Platte ausfallen, sind alle Daten verloren.

- **RAID-6 (Parity Striping):** RAID-6 funktioniert wie RAID-5, ist aber doppelt redundant und erfordert zumindest vier Festplatten. Selbst beim Ausfall von zwei Festplatten kommt es zu keinem Datenverlust.

Zusätzlich zu den aktiven RAID-Festplatten können Sie eine oder mehrere Reserve-Festplatten (*spare devices*) einrichten, was aber unüblich (und bei RAID-0 unmöglich) ist. Reserve-Geräte werden in diesem Buch nicht weiter behandelt.

Hardware- versus Software-RAID

RAID kann auf drei Arten realisiert werden: per Hardware durch einen RAID-Controller, per Software über das BIOS in Kombination mit einem Festplatten-Controller oder per Software durch das Betriebssystem (z. B. Windows oder Linux). Jede Variante hat Vor- und Nachteile:

- **Hardware-RAID:** Beim Hardware-RAID kümmert sich ein RAID-Controller um die Verwaltung der Festplatten. Der Rechner (die CPU) wird dadurch nicht belastet. Echtes Hardware-RAID ist im Regelfall schnell, stabil und unterstützt teilweise sogar den Austausch defekter Platten im laufenden Betrieb (*hot swap*). Allerdings sind richtige Hardware-RAID-Controller teuer und nicht immer Linux-kompatibel (d. h., für viele »echte« RAID-Controller fehlen geeignete Linux-Treiber).
- **BIOS-Software-RAID (Fakeraid):** Beim BIOS-Software-RAID realisiert das BIOS eines ansonsten recht gewöhnlichen (also billigen) Festplatten-Controllers in Kombination mit einem Betriebssystemtreiber verschiedene RAID-Level. Im Vergleich zu Hardware-RAID hat Software-RAID den Nachteil, dass sich statt eines eigenen Controllers die CPU um RAID-Details kümmern muss. Das ist ein Geschwindigkeitsnachteil. Der Begriff *Fakeraid* erklärt sich daraus, dass viele BIOS-RAID-Controller so angepriesen werden, als wären sie echte Hardware-RAID-Controller – und das ist ganz einfach falsch. Das gilt insbesondere für nahezu alle gängigen SATA-RAID-Lösungen:

<http://linux-ata.org/faq-sata-raid.html>

- **Linux-Software-RAID:** Moderne Betriebssysteme bieten die Möglichkeit, einige RAID-Level selbst zu realisieren. Je nach RAID-Level kann die gleiche Datenübertragungsrate wie mit einem Hardware-RAID erreicht werden, allerdings bei einer höheren CPU-Belastung.

Unter Linux ist für ein Software-RAID der sogenannte *Multi Devices Driver Support* `md_mod` zuständig. `md_mod` bildet aus mehreren Festplatten-Partitionen ein neues Device (`/dev/mdn`), auf das der Dateisystemtreiber zugreifen kann. Nach der RAID-Konfiguration verwenden Sie nicht mehr direkt eine Festplattenpartition, sondern eine RAID-Partition `/dev/mdn`, um darauf Ihr Dateisystem einzurichten.

Ein Linux-kompatibler Hardware-RAID-Controller, der vom Ubuntu-Installationsprogramm unterstützt wird, ist sicherlich die optimale Lösung. Mangels eines geeigneten Controllers habe ich persönlich damit aber leider keine Erfahrung.

In allen anderen Fällen lautet meine Empfehlung: Verwenden Sie ein Linux-Software-RAID-1 (zwei Festplatten) oder -RAID-5 (drei oder mehr Festplatten)! Diese Varianten werden auch bei der Installation von Ubuntu Linux ausgezeichnet unterstützt.

Vorkonfigurierte Rechner mit BIOS-RAID

Wenn Sie einen vorkonfigurierten (Windows-)Rechner mit mehreren Festplatten kaufen, sind diese oft schon als BIOS-RAID-Verbund vorkonfiguriert. Sofern Sie auf das installierte Windows verzichten können, öffnen Sie beim Rechnerstart die Konfigurationsdialoge des BIOS und deaktivieren dort das RAID. Anschließend können Sie Linux ganz normal installieren – je nach Wunsch ganz ohne RAID oder mit einem Linux-Software-RAID. Allerdings verlieren Sie bei dieser Vorgehensweise alle Daten, die sich bisher auf den Festplatten befunden haben.

Wesentlich schwieriger ist es, die Windows-Installation zu erhalten (ein Wunsch, der bei einer Server-Installationen zum Glück selten aufkommt): Das Problem besteht darin, dass Linux mit `dmraid` zwar einen Treiber für nahezu alle gängigen BIOS-RAID-Controller hat (je nach Controller nur für die RAID-Level 0 und 1), dass dieser Treiber während der Installation aber nicht oder nur schlecht unterstützt wird. Wenn Sie nicht aufpassen, können Sie sogar die vorhandene Installation irrtümlich zerstören.

Ein Ausweg besteht darin, eine zusätzliche Festplatte zu kaufen und dort die Boot- und Systempartition des Servers einzurichten. Wenn dieses System einmal läuft, installieren Sie das `dmraid`-Paket und können dann im vorhandenen BIOS-RAID-Verbund weitere Linux-Partitionen anlegen.

Echte Linux-Freaks können Ubuntu halb-manuell auch direkt in einen BIOS-RAID-Verbund installieren. Zwei entsprechende Anleitungen finden Sie hier:

<https://help.ubuntu.com/community/FakeRaidHowto>
<http://ubuntuforums.org/showthread.php?t=464758>

GRUB und RAID

Im Folgenden gehe ich in diesem Buch nur noch auf Linux-Software-RAIDs ein. »RAID« meint also von nun an Linux-Software-RAID!

Die aktuelle Version des Linux-Bootloaders GRUB (0.97 in Ubuntu 8.04) unterstützt RAID eigentlich nicht. Eine inoffizielle und nicht dokumentierte Ausnahme ist RAID-1. Der Ubuntu-Installer unterstützt diese Variante, installiert in diesem Fall GRUB allerdings nur in den Bootsektor der ersten Festplatte. Fällt diese Festplatte aus, enthält die zweite RAID-1-Festplatte dank *mirroring* zwar die zum Start erforderlichen Daten, GRUB selbst kann aber nicht gestartet werden. Abhilfe schafft die manuelle Installation von GRUB in die Bootsektoren aller Festplatten, die den RAID-1-Verbund bilden (siehe Seite 155).

Sie können GRUB auch einfach in einer eigenen Bootpartition ohne RAID anlegen. Das vereinfacht die Installation, führt aber dazu, dass Ihr Server nach dem Ausfall einer Festplatte nicht mehr startet. RAID sichert in diesem Fall zwar Ihre Daten, es ist aber Handarbeit erforderlich, um den Rechner nach dem Austausch der defekten Festplatte neu zu starten und GRUB wieder einzurichten.

Statt GRUB können Sie auch den älteren Boot-Loader LILO verwenden, der RAID-0-kompatibel ist. Ich empfehle den Einsatz von LILO aber nicht – das Programm ist nicht mehr zeitgemäß und relativ störungsanfällig.

Swap-Partitionen und RAID

Aus Geschwindigkeitsgründen ist es nicht empfehlenswert, Swap-Daten in einer RAID-Partition anzulegen – selbst dann nicht, wenn Sie RAID-0 verwenden. Am effizientesten ist es, einfach auf jeder Festplatte eine ganz gewöhnliche Partition als Swap-Speicher vorzusehen. Linux parallelisiert den Zugriff auf den Swap-Speicher selbstständig und braucht hierfür keinen RAID-Verbund.

Wenn Ihnen Stabilität wichtiger als Geschwindigkeit ist, empfiehlt es sich dennoch, für den Swap-Speicher eine RAID-1- oder RAID-5-Partition vorzusehen. Der Grund: Wenn im laufenden Betrieb eine Festplatte ausfällt, können auch Daten im Swap-Bereich verloren gehen. Das kann zum Absturz des Rechners führen. Der Sicherheitsgewinn wird durch einen deutlich langsameren Zugriff auf den Swap-Speicher erkaufte.

RAID-Speicher vergrößern, RAID-Level ändern etc.

Es ist grundsätzlich nicht möglich, einen vorhandenen RAID-Verbund zu erweitern oder den RAID-Level eines Verbunds zu ändern (also z. B. von RAID-1 auf RAID-5 nach dem Kauf der dritten Festplatte)! Derartige Operationen erfordern jeweils ein komplettes Backup der Daten (z. B. auf einer weiteren, externen Festplatte) und eine aufwendige Neukonfiguration bzw. Neuinstallation. RAID-Systeme sind also inflexibel, und es lohnt sich, das System von vornherein gut zu planen und großzügig zu dimensionieren.

Ein Sonderfall ist die Umwandlung einer herkömmlichen Installation auf einer Festplatte in ein RAID-1-System mit zwei Festplatten. Wie das gelingt, verrät eine genaue Anleitung ab Seite 162.

1.5 Logical Volume Manager (LVM)

Der *Logical Volume Manager* setzt eine logische Schicht zwischen das Dateisystem und die Partitionen der Festplatte. Linux-intern ist dafür das Kernelmodul `dm_mod` zuständig. LVM kann auch RAID-Partitionen nutzen, d. h., Sie können zuerst aus mehreren physikalischen Festplatten-Partitionen einen RAID-Verbund bilden und darin dann LVM einrichten. Dieser Abschnitt gibt eine Einführung in die Grundlagen von LVM. Details zur Installation und zur späteren Administration eines LVM-Systems folgen in den Kapiteln 2 und 8. Weitere Informationen finden Sie hier:

http://de.wikipedia.org/wiki/Logical_Volume_Manager
<http://sourceware.org/lvm2/>

Vor- und Nachteile

Auch wenn die einleitende Beschreibung des LVM-Prinzips abstrakt klingt, sind die Vorteile in der Praxis sehr real und leicht verständlich:

- Im Rahmen des von LVM verwalteten Festplattenbereichs können Sie im laufenden Betrieb ohne Rechnerneustart Partitionen anlegen, vergrößern und verkleinern. Den vorhandenen LVM-Speicherpool können Sie jederzeit durch den Einbau einer weiteren Festplatte vergrößern.
- Sie können dank LVM Bereiche mehrerer Festplatten zu einer einzigen, riesigen virtuellen Partition zusammenfassen.
- Sie können sehr einfach einen sogenannten Snapshot eines Dateisystems erstellen. Das ist ideal für Backups im laufenden Betrieb.
- LVM ist sehr schnell. Sie bezahlen für die höhere Flexibilität also nicht mit einer spürbar verringerten Geschwindigkeit. (Der Geschwindigkeitsunterschied gegenüber dem direkten Ansprechen einer Festplattenpartition ist kaum messbar. Die CPU-Belastung ist nur geringfügig höher.)

Natürlich gibt es auch Nachteile:

- LVM kann nicht dazu verwendet werden, um eine herkömmliche Partition zu vergrößern, die ohne LVM erzeugt wurde. (Sie können LVM also nicht erst dann einsetzen, wenn Ihre herkömmlichen Partitionen schon voll sind).
- LVM kümmert sich nur um Partitionen, nicht um die darauf enthaltenen Dateisysteme. Bei einer Verkleinerung müssen Sie zuerst das Dateisystem verkleinern, dann die LVM-Partition. Bei einer Vergrößerung müssen Sie zuerst die LVM-Partition vergrößern, dann das Dateisystem.
- Auch wenn LVM bei der Verwaltung von Partitionen mehr Flexibilität gestattet, macht es doch die Administration als Ganzes komplizierter. Das gilt nicht zuletzt deswegen, weil die LVM-Nomenklatur ein wenig verwirrend ist (*physical volumes, volume groups, logical volumes* etc.).
- Aktuelle GRUB-Versionen sind nicht in der Lage, Linux von einer LVM-Partition zu starten. Sie brauchen also eine `/boot`-Partition außerhalb des LVM-Bereichs oder das in Entwicklung befindliche GRUB 2, das voraussichtlich 2010 fertiggestellt wird.

Glossar

Die Fülle ähnlich lautender Begriffe und Abkürzungen erschwert den Einstieg in die LVM-Welt. Um die Konfusion nicht noch zu vergrößern, verzichte ich in diesem Abschnitt bewusst auf eine Übersetzung der Begriffe.

Zwischen der Festplatte und dem Dateisystem stehen drei Ebenen: *physical volumes, volume groups* und *logical volumes*:

- **Physical volume (PV):** Ein PV ist im Regelfall eine von LVM verwaltete Partition der Festplatte. Es kann sich auch um eine ganze Festplatte oder um ein RAID-Device handeln. Entscheidend ist, dass die Partition, die Festplatte oder der RAID-Verbund als PV gekennzeichnet ist, damit die unterschiedlichen LVM-Kommandos funktionieren.
- **Volume group (VG):** Ein oder mehrere *physical volumes* können zu einer Gruppe zusammengefasst werden. Auf diese Weise ist es möglich, Partitionen unterschiedlicher Festplatten quasi zusammenzuhängen, also einheitlich zu nutzen. Die *volume group* stellt eine Art Speicherpool dar, der alle zur Verfügung stehenden physikalischen Speichermedien vereint. Dieser Pool kann jederzeit um weitere *physical volumes* erweitert werden.
- **Logical volume (LV):** Ein *logical volume* ist ein Teil der *volume group*. Für den Anwender wirkt ein *logical volume* wie eine virtuelle Partition. Im *logical volume* wird das Dateisystem angelegt. (Das heißt, anstatt ein Dateisystem in `/dev/sda7` anzulegen, geben Sie jetzt den Device-Namen des *logical volume* an.) Falls in der *volume group* noch Speicher verfügbar ist, können *logical volumes* jederzeit vergrößert werden.

Zur Veranschaulichung der drei Begriffe folgt hier ein Beispiel (siehe Abbildung 1.2): Auf einem System dienen die drei RAID-1-Partitionen `/dev/md0`, `/dev/md1` und `/dev/md2` als *physical volumes* für eine *volume group* eines LVM-Systems. `/dev/md0` umfasst 400 GByte, `/dev/md1` und `/dev/md2` umfassen jeweils 500 GByte. Der LVM-Speicherpool (also die *volume group*) ist somit 1,4 TByte groß. Darin befinden sich nun diverse *logical volumes*:

- LV1 mit der Systempartition (50 GByte)
- LV2 mit der Partition `/var` (200 GByte)
- LV3 mit der Partition `/var/www` (200 GByte)
- LV4 mit der Partition `/home` (400 GByte)

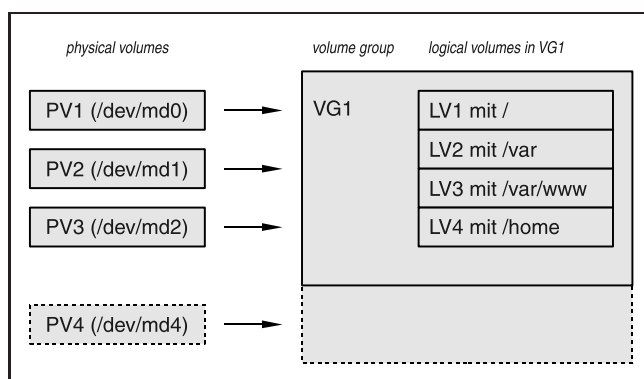


Abbildung 1.2: LVM-System

1.5 Logical Volume Manager (LVM)

Insgesamt sind somit 850 GByte mit Partitionen belegt, und 550 GByte sind noch frei. Damit können Sie zu einem späteren Zeitpunkt vorhandene Partitionen vergrößern oder neue Partitionen anlegen. Sollte der gesamte LVM-Pool erschöpft sein, können vorhandene LVs/Dateisysteme verkleinert werden (wenn sich herausgestellt hat, dass sie ursprünglich zu großzügig dimensioniert wurden), um so Platz zur Vergrößerung anderer LVs/Dateisysteme zu schaffen. Reicht das nicht aus, fügen Sie zwei weitere Festplatten hinzu, konfigurieren diese wieder mit RAID-1 und fügen die RAID-1-Partition als viertes *physical volume* zur *volume group* hinzu.

In der LVM-Dokumentation kommen noch zwei weitere Begriffe häufig vor:

- **Physical device (PD):** Dabei handelt es sich ganz einfach um eine Festplatte. LVM kann wahlweise die gesamte Festplatte oder auch mehrere Partitionen dieser Festplatte in Form von *physical volumes* nutzen.
- **Physical extent (PE):** Bei *volume groups* und *logical volumes* kann nicht jedes einzelne Byte einzeln verwaltet werden. Die kleinste Dateneinheit ist vielmehr ein *physical extent* (standardmäßig 4 MByte). Die Anzahl der PEs ist unbegrenzt. Zu viele PEs machen aber die Verwaltung ineffizient, weswegen Sie für sehr große *logical volumes* die Größe von PEs hinaufsetzen sollten.

Um es kurz zusammenzufassen: Eine minimale LVM-Konfiguration sieht so aus, dass Sie eine oder mehrere Festplatten- oder RAID-Partitionen für den Einsatz von LVM kennzeichnen. Das sind die *physical volumes* Ihres LVM-Systems. Anschließend definieren Sie eine *volume group* und fügen dort alle Partitionen ein. Damit haben Sie nun Ihren LVM-Speicherpool. Darin können Sie beliebig viele *logical volumes* definieren, also gleichsam virtuelle Partitionen, in denen Sie dann ein Dateisystem einrichten. Bei allen hier aufgezählten Schritten unterstützt Sie das Ubuntu-Installationsprogramm, das im nächsten Kapitel im Detail beschrieben wird.

GRUB und LVM

Aktuelle GRUB-Versionen sind nicht in der Lage, Linux von einer LVM-Partition zu starten. Die einfachste Lösung besteht darin, eine Bootpartition außerhalb des LVM-Bereichs anzulegen. Andere Alternativen sind LILO, das bereits jetzt LVM-kompatibel ist, oder GRUB 2, das vermutlich 2010 die aktuelle GRUB-Version 0.97 ablösen wird.

Swap-Partitionen und LVM

Es ist nicht zweckmäßig, Swap-Partitionen innerhalb eines LVM-Bereichs anzulegen. Der Grund: Bei LVM kann es passieren, dass aufeinanderfolgende Datenblöcke einer Partition auf ganz unterschiedlichen Orten der Festplatte(n) landen. Das macht den Zugriff auf LVM-Swap-Partition ineffizient. (Der Ubuntu-Installer hält sich übrigens nicht an diese Regel: Die Partitionierungsmethode GEFÜHRT – GESAMTE PLATTE VERWENDEN UND LVM EINRICHTEN richtet die Swap-Partition in einem *logical volume* ein.)

1.6 Konfigurationsempfehlungen

Nachdem Sie nun die wichtigsten Parameter einer Server-Installation kennen, gibt der letzte Abschnitt dieses Kapitels einige konkrete Installationsempfehlungen. Vorweg einige Grundregeln:

- **KISS:** Diese aus dem Englischen kommende Abkürzung für *Keep it simple, stupid!* (wörtlich: Mach's einfach, du Dummkopf!) bedeutet für eine Server-Installation: Vermeiden Sie unnötig komplizierte Lösungen. Überschaubare, einfache Ansätze sind fast immer besser als ein irregeleiteter Perfektionsanspruch. Das KISS-Prinzip erleichtert die Wartung, verkürzt die Fehlersuche und gibt Ihnen die Möglichkeit, Aufgaben zu delegieren.
- **LTS:** Es gibt halbjährlich neue Ubuntu-Versionen, aber nur alle zwei Jahre eine LTS-Variante (*long term support*) – zuletzt im April 2008 (Ubuntu 8.04). Bei den LTS-Versionen werden alle Server-Pakete fünf Jahre lang gewartet und bei Sicherheitsproblemen aktualisiert – und das kostenlos! Bei gewöhnlichen Ubuntu-Versionen beträgt der Wartungszeitraum nur 18 Monate.

Verwenden Sie unbedingt die letzte LTS-Version, auch wenn diese älter ist als die gerade aktuelle Ubuntu-Version! Ein Server-Update bzw. eine Neuinstallation ist mit großem Aufwand verbunden. Sobald die Konfiguration einmal abgeschlossen ist, wird Ihr Server klaglos laufen. Mit einer LTS-Version verlängern Sie die Laufzeit Ihres Servers ganz erheblich und sparen so Zeit und Nerven.

- **64-Bit:** Obwohl ich noch immer gewisse Zweifel hege, ob 64-Bit-Installationen im Desktop-Bereich sinnvoll sind, stellt sich im Server-Segment die Frage nach dem Für und Wider eines größeren Adressraums erst gar nicht: Verwenden Sie die 64-Bit-Variante von Ubuntu!

Auch wenn Ihr Server momentan nur mit ein oder zwei GByte RAM bestückt ist: Ein RAM-Ausbau ist einfach, billig und kann je nach Anwendung eine gewaltige Leistungsverbesserung mit sich bringen – aber eben nur dann, wenn Sie die 64-Bit-Version einsetzen.

- **(Daten-)Sicherheit ist wichtiger als Performance:** Bei der Nutzung von Festplatten haben Sie oft die Wahl zwischen etwas schnelleren und etwas sicheren Lösungen. Bei den meisten Servern sollten Datensicherheit, Stabilität und hohe Verfügbarkeit Vorrang haben. Bedenken Sie nicht nur Ihre Backup-Strategie, sondern auch, wie lange Ihr Server beim Ausfall einer Festplatte *offline* ist.

Und machen Sie sich keine Illusionen: *Jede* Festplatte fällt irgendwann aus – manche früher, andere später ...

Zu jeder dieser Grundregeln lassen sich gut begründete Ausnahmen finden. Wer schon den zehnten oder hunderten Server konfiguriert, weiß es aus eigener Erfahrung sowieso besser ... Aber wenn Sie erst am Beginn Ihrer Laufbahn als Server-Administrator stehen, machen Sie nichts falsch, wenn Sie diese Regeln befolgen.

1.6 Konfigurationsempfehlungen

Server-Installation auf eine Festplatte

Die Installation auf einer Festplatte ist sicherlich die einfachste Variante: Da RAID ausscheidet, bleibt nur noch die Überlegung, ob Sie LVM einsetzen oder nicht. Argumente dafür sind die einfachere Erweiterbarkeit, die einfache Installation (der Ubuntu-Installer erledigt die ganze Arbeit nahezu allein) sowie die höhere Flexibilität, einzelne Datenpartitionen nachträglich zu verkleinern bzw. zu vergrößern. Argumente dagegen gibt es kaum – sieht man einmal von der Grundidee des KISS-Konzepts ab.



Hinweis

Abbildung 1.3 und alle weiteren Abbildungen dieses Abschnitts geben nur den Namen der Device-Datei an, nicht den vollständigen Pfad (also sda1 statt /dev/sda1). Pro Festplatte gibt es maximal drei physikalische Partitionen, die alle primär sind. Es besteht also keine Notwendigkeit, zwischen verschiedenen Partitionstypen zu unterscheiden.

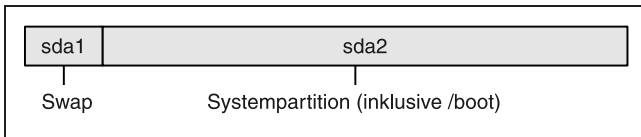


Abbildung 1.3: Minimalistische Konfiguration ohne LVM und mit nur einer Systempartition

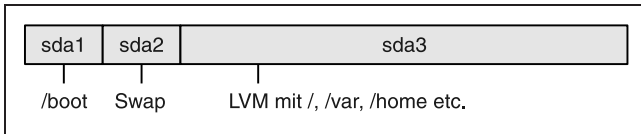


Abbildung 1.4: Konfiguration mit LVM und mehreren Datenpartitionen

Eine spätere Vergrößerung des Datenvolumens ist bei der LVM-Variante kein Problem: Bauen Sie eine zweite Festplatte ein, und fügen Sie die ganze Festplatte als *physical volume* zur bereits vorhandenen *volume group* hinzu. Wesentlich schwieriger ist es, das System zu einem späteren Zeitpunkt auf RAID-1 umzustellen (siehe Seite 162).

Server-Installation auf zwei Festplatten

Zwei Festplatten sind für mich der ideale Ausgangspunkt für einen Server, solange nicht riesige Datenmengen im Spiel sind. Wenn Sie Wert auf Daten- und Ausfallsicherheit legen, führen zwei Festplatten zwangsläufig zu RAID-1. Die einzig offene Frage ist, ob Sie mit oder ohne LVM arbeiten möchten. Abbildung 1.5 und Abbildung 1.6 enthalten zwei Installationsvorschläge.

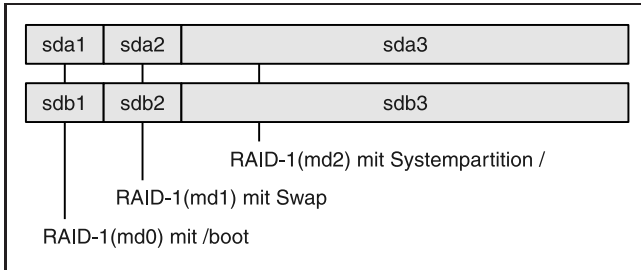


Abbildung 1.5: RAID-1 auf zwei Festplatten

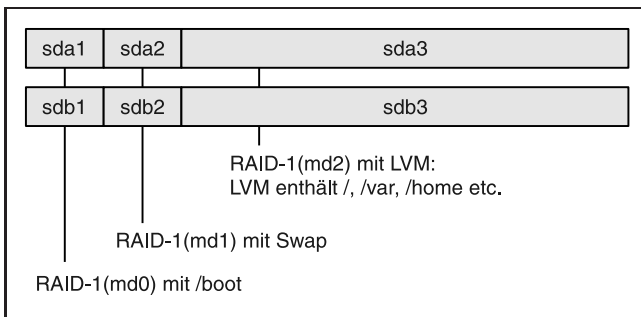


Abbildung 1.6: RAID-1 und LVM auf zwei Festplatten

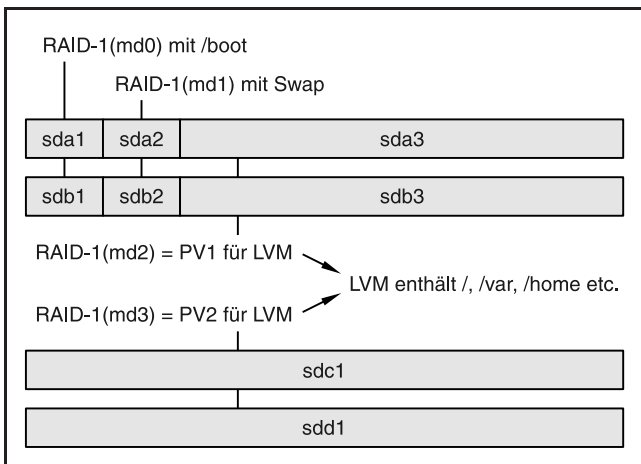


Abbildung 1.7: Nachträgliche Erweiterung der Konfiguration aus Abbildung 1.6

Eine spätere Vergrößerung des Datenvolumens ist ohne LVM nahezu unmöglich. Wenn Sie sich für die LVM-Variante entschieden haben, erweitern Sie Ihr System am besten um zwei Festplatten, die Sie zu einem RAID-1-Verbund kombinieren. Die resultierende RAID-Partition fügen Sie als *physical volume* zur bereits vorhandenen *volume group* für die System- und Datenpartitionen hinzu (siehe Abbildung 1.7).

1.6 Konfigurationsempfehlungen

Server-Installation auf drei oder mehr Festplatten

Eines vorweg: Ich bin kein Freund vieler Festplatten. Mit jeder Festplatte steigen die Fehlerwahrscheinlichkeit, der Stromverbrauch, das Abwärmeproblem und der Administrationsaufwand. Verwenden Sie lieber wenige, große Festplatten als viele kleine (auch wenn das die anfänglichen Investitionskosten ein wenig vergrößert). Ich gebe schon zu: Bei großen Servern haben Sie keine andere Wahl: Sie werden die größten verfügbaren Festplatten einsetzen, und davon so viele wie möglich ...

Der naheliegende RAID-Level 5 (siehe Abbildung 1.8) bereitet mir Bauchweh, wenn er ohne einen Hardware-RAID-Controller betrieben wird: Die Datensicherheit ist zwar gut, aber beim Ausfall einer Platte steht das System für viele Stunden, die zur Initialisierung der Ersatzplatte erforderlich sind.

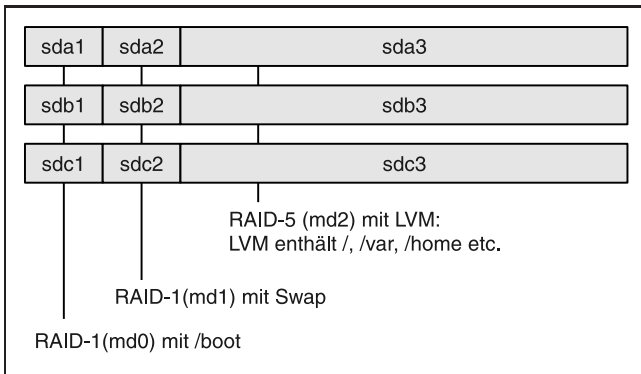


Abbildung 1.8: RAID-5 und LVM auf drei oder mehr Festplatten

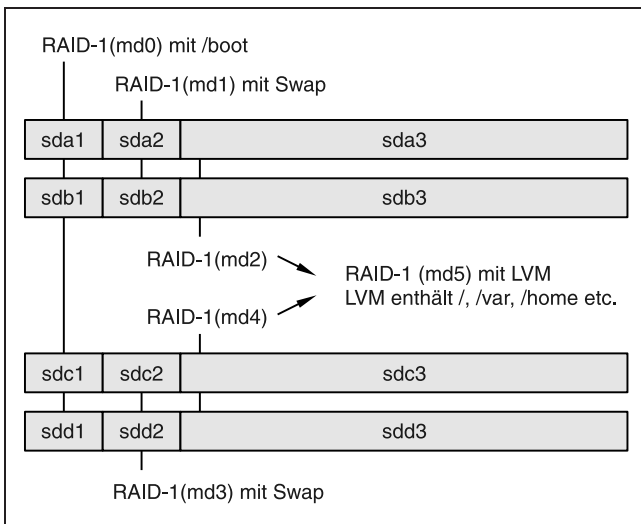


Abbildung 1.9: RAID-10 und LVM auf vier Festplatten

Bei vier Festplatten ist RAID-10 eine interessante Option (siehe Abbildung 1.9): Zwar bleibt nur die Hälfte der gesamten Festplattenkapazität für die Daten übrig, dafür ist das System gegenüber der RAID-5-Konfiguration deutlich schneller und lässt sich im Fall eines Defekts rascher wieder nutzen.

Beachten Sie, dass die Boot- und Swap-Partition(en) jeweils eine RAID-1-Partition nutzen, wenn auch aus unterschiedlichen Gründen: Bei der Bootpartition deswegen wird deswegen RAID-1 verwendet, weil GRUB inkompatibel zu anderen RAID-Leveln ist; bei der Swap-Partition kommt es zum Einsatz, weil RAID-10 keinen Geschwindigkeitsgewinn mit sich bringen würde.