Bachelorarbeit

Dokumentation zum CGA-Backup

Institut für Computergraphik und Algorithmen
der
Technischen Universität Wien

Stephan Plepelits, 9626338
plepelits@cg.tuwien.ac.at

August 2008

Inhaltsverzeichnis

Einleitung

Backups von Daten anzulegen ist eine der wichtigsten Möglichkeiten um Datenverlust zu vermeiden. Ein RAID, also einen Verbund aus mehreren Platten, zu verwenden auf dem die Daten redundant gespeichert werden, ist schon eine gute Sache, allerdings können auch hier unerwartete Dinge passieren. Das System kann crashen und alle Daten mitreißen, oder der Controller eingehen, wo zumindest längere Wartezeiten entstehen, bis das System wieder lauffähig ist. Sehr oft passiert es auch, dass Dateien und Verzeichnisse absichtlich oder unabsichtlich gelöscht werden, diese dann aber wieder gebraucht werden, was aber oft bereits einige Zeit zurückliegen kann. In dem vorliegenden Dokument wird gezeigt, wie am Institut für Computergraphik und Algorithmen der TU-Wien dieses Problem gelöst wurde.

Ein geschichtlicher Abriss

Natürlich sind Backups keine neue Idee. All erdings haben sich die Methoden und Mittel über die Jahre stark geändert und waren wie alles in der Computertechnik vom technologischen Fortschritt beeinflusst. Ende der 1990er Jahre wurden am Institut die Backups noch auf Band gespeichert, was zwar damals relativ kostengünstig war, dafür aber arbeitsaufwendig (es mussten jeden Tag die Bänder getauscht werden) und unflexibel (die Bänder waren nicht besonders groß, der Zugriff war langwierig).

Nachdem die Festplattengrößen in den 1990er Jahren stark gewachsen und die Preise stark gesunken sind, wurde die Idee geboren, auf Festplatten zu backupen. Vorteile sollten sein, dass die Backups wesentlich wartungsfreier ablaufen, da nicht jeden Tag Bänder getauscht werden müssen und dass diese Backups über einen größeren Zeitraum gespeichert werden können. Um eine höhere Sicherheit zu haben wurde der Backup-Server bei einem anderen Institut aufgestellt. Damit sollten die Daten auch rekonstruierbar sein, sollte das Institut Opfer einer Katastrophe werden, z.B. eines Brandes.

Die erste Version hat „tar“ benützt um die Backups zu packen und zu übertragen, je nach Wochentag als volle oder inkrementelle Backups (in dem sich nur geänderte Dateien befanden). Da bei „tar“ die Daten in einer Datei gespeichert werden, war es sehr mühsam dort wieder einzelne Dateien zu suchen geschweige denn zu extrahierern.

Dies war dann auch Hauptgrund für eine neue Backupstrategie. Im Rahmen einer Institutsklausur wurde ein Anforderungsprofil erstellt, dass auf folgende Ideen basierte:

- Die Backups sollen auf längere Zeit zurück gespeichert werden
- Die Backups sollen auf einem RAID gespeichert werden
- Die Backups sollen auch für die InstitutsmitarbeiterInnen zugänglich sein

Daraufhin wurde eine neues Backup entworfen, dass auf „rsync“ basiert. Rsync dupliziert einen Verzeichnisbaum an einen anderen Speicherort, vergleicht aber die Daten mit dem Zielort und kopiert nur die veränderten Dateien. Mächtig wird die Lösung dadurch, dass Dateien, die in mehreren Backupinstanzen vorkommen (was den Großteil aller Dateien ausmacht), nur einmal im Dateisystem gespeichert werden.

Es kamen immer wieder andere Probleme auf, z.B. war die Festplatten mit der Zeit zu voll geworden und es mussten immer wieder Daten zwischen den Platten hin- und hergeschoben werden, was zu einer unübersichtlichen Struktur geführt hat und auch immer wieder zu Datenverlusten. Dies wurde durch die Einführung eines Logical Volume Managers gelöst.

Ein anderes Problem war, dass die Backups aufteilt waren auf das aktuelle vollständige Backup, einer monatlichen Kopie des vollständigen Backups und nichtvollständigen Unterschieden zwischen den einzelnen Backups. Dies wurde endgültig dadurch gelöst, dass es nur noch vollständige Backups gibt, die mit Hilfe einer gewissen Strategie wieder aus dem Dateisystem gelöscht werden.

Verwendete Technologien

GNU/Linux

Der Backupserver läuft mit einer aus Sicherheitsgründen auf die notwendigen Dienste beschränkte Installation von Debian GNU/Linux 4.0. UNIX-Varianten, und auch Linux als verwandtes System, eignen sich hervorragend für den Einsatz als Server-Betriebssysteme, da sie sehr einfach aus der Ferne zu bedienen sind und der Zugriff über eine Shell einfach und schnell möglich ist. Hier eine Liste der Programme und Funktionen, die für das Backup verwendet werden:

- Perl

  Perl ist eine sehr weit verbreitete Skriptsprache, die als vollständige Programmiersprache geeignet ist. Alle Skripte rund um das Backup-System (mit Ausnahme der Webpage) sind in Perl geschrieben.

- Cron

  Cron ist ein Dienst unter Linux, mit dessen Hilfe der automatische Aufruf von Skripten und Programmen festgelegt werden kann. Die Aufrufzeit ist sehr flexibel festlegbar. Auf der Client-Seite (also den Rechnern die gebackupt werden) wird die Ausführung der Backups damit angestossen, Server-seitig werden Skripte gestartet um alte Backups zu löschen bzw. um Statistiken zu berechnen.

- RSH/SSH

  RSH (Remote Shell) bzw. SSH (Secure Shell) sind Programme um auf entfernten Rechnern Zugang zu bekommen. Dies wird nicht nur in der alltäglichen Arbeit verwendet um die Server zu administrieren, auch werden diese verwendet um die Daten der Backups zu übertragen. Der Nachteil davon ist, dass es notwendig ist am Backupserver einen Shell-Zugang zu haben, was eine gewissen Unsicherheit in das Backup bringt. An einer Alternative dafür wird daher gearbeitet.

- EXT3

  EXT3 ist ein sehr beliebtes und sicheres Dateisystem, in dem die Daten der Backups abgelegt werden. Es verfügt über ein Rechtesystem in dem für jede Datei und jedes Verzeichnis Eigentümer und Gruppenzugehörigkeit festgelegt werden können für diese die Zugriffsrechte Lesen (r), Schreiben (w) und Ausführen (x).

  In UNIX-Betriebssystem wird die Dateiinformation in einem sogenannten Inode gespeichert, der einen Verweis auf den Dateiinhalt enthält. Diese Inodes können vervielfacht werden um eine Datei in mehreren Verzeichnissen zu referenzieren, was als harter Verweise (hard link) bezeichnet wird. Das Dateisystem achtet darauf, dass der Dateiinhalt erst freigegeben wird, sobald alle Inodes entfernt sind. Harte Verweise sind nicht auf ein anderes Dateisystem möglich.

  Diese harten Verweise werden zwischen einzelnen Backupinstanzen genutzt, damit Dateien die sich nicht geändert haben nur einmal gespeichert werden müssen.

  Mit einem einfachen Befehl kann eine Kopie des Verzeichnisbaumes angelegt werden, der alle Dateien als hard Links beinhält. Diese Kopie des Verzeichnisbaum braucht nur minimalen zusätzlichen Speicher im Dateisystem.

  Weiters sind noch die symbolischen Verweise erwähnenswert. Dies sind einfache Verweise auf andere Dateien oder Verzeichnisse und können im Gegensatz zu harten Verweisen auch auf ein anderes Dateisystem zeigen.

  Das EXT3-Dateisystem ermöglicht es auch, dass das Dateisystem mit der Zeit vergrößert wird. Die Vergrößerung kann sogar durchgeführt werden, während es in Benutzung ist. Eine Verkleinerung ist prinzipiell auch möglich, allerdings darf das Dateisystem zu dieser Zeit nicht eingehängt sein.

- Suid

  Suid (Set User ID) ist eine Methode um einzelne Programmteile mit anderen Berechtigungen auszuführen als den Rest. Bei UNIX-Betriebssystemen ist dies ein Flag im Dateisystem, der dazu führt, dass diese Programme mit den Rechten des Dateiinhabers ausgeführt werden im Gegensatz zu den Rechten des Elternprozesses.

RAID und LVM

RAID

Ein RAID (Redundant Array of Inexpensive/Individual Disks) ist ein Verbund mehrerer Festplatten, die im System als einzelne Festplatte verwendet werden. Diese Methode soll vor allem zwei Ansprüche befriedigen: Größerer durchgehender Festplattenplatz und höhere Ausfallssicherheit. Manchmal wird auch noch größerer Datendurchsatz erwartet, dies war aber für das Backup kein ausschlaggebendes Argument.

Unterscheiden muss man noch zwischen einem Software-RAID und einem Hardware-RAID. Ein Software-RAID implementiert die Funktionalität des RAIDs im Betriebssystem, der Computer braucht also keine besonderen Bauteile. Ein Hardware-RAID dagegen basiert auf einem eigenen Baustein im Computer, der sich ausschließlich um das RAID kümmert, dem Betriebssystem wird vorgegaukelt, dass es sich um eine physikale Festplatte handelt. Da diese Bausteine allerdings sehr teuer sind haben sich inzwischen Zwischenlösungen entwickelt. Für das Backup waren die Vorteile die ein Hardware-RAID bietet nicht ausschlaggebend, darum haben wir uns für ein Software-RAID entschieden, welches unter Linux sehr stabil funktioniert.

Je nach Anforderung gibt es verschiedene Arten von RAIDs, die sogenannten „Levels“. Hier die wichtigsten:

- RAID 0

  Bei einem RAID 0 ist das Ziel ein größeres Datenvolumen zu haben. Dazu werden zwei oder mehr Platten zusammengefügt und die Daten verteilt geschrieben, was auch höheren Datendurchsatz möglich macht. Es werden allerdings keine Sicherheitsinformationen geschrieben, sobald eine Festplatte ausfällt sind alle Daten weg.

- RAID 1

  Bei einem RAID 1 werden die Daten parallel auf zwei gleich großen Laufwerken gespeichert. Wenn also eines diese Laufwerke ausfällt, sind die Daten noch alle vorhanden. Allerdings wird mit dem Platz recht verschwenderisch umgegangen, werden doch einfach alle Daten dupliziert. Unter Linux wird dies vor allem für die Systemplatte verwendet, da dies das einzige Software-RAID-Level ist, von dem auch gebootet werden kann.

- RAID 5

  Ein RAID 5 speichert keine Kopie der Daten sondern sogenannte Parity-Informationen verteilt auf die Laufwerke. Es bietet also n-1 mal den Platz eines einzelnen Laufwerks, wobei diese gleich groß sein sollten (sonst wird der Platz des kleinsten Laufwerks angenommen). Dieses System ist beliebt, da es viel Platz bei wenigen Laufwerken bietet.

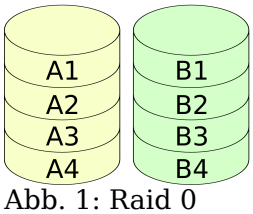

Abb. 1: Raid 0

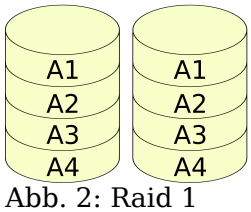

Abb. 2: Raid 1

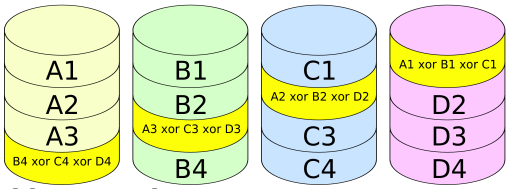

Abb. 3: Raid 5

LVM

In einem normalen Computersystem ist die Festplatte meist in mehrere getrennte Bereiche, sogenannte Partitionen, geteilt. Eine nachträgliche Änderung dieser Partitionen ist allerdings eine mühsame Angelegenheit, die auch oft schon zu Datenverlusten geführt hat. Ein großer Nachteil ist, dass eine Änderung der Partitionierung normalerweise auch einen Reboot notwendig macht.

Um diesem Problem entgegen zu wirken wurde der LVM (Logical Volume Manager) entwickelt. Dieser teilt den Festplattenplatz ähnlich den Partitionen in verschiedene Teile, allerdings können diese sehr einfach verkleinert und vergrößert werden. Ein LVM ermöglicht es auch, dass sich Partitionen über mehrere Festplatten erstrecken. Logisch wird dies so geregelt, dass es mehrere physikale Geräte (physical devices) gibt, die in einer Laufwerkgruppe verwendet werden (volume group). In jeder dieser Laufwerkgruppen können dann mehrere logische Laufwerke (logical volumes) angelegt werden.

Die Festplatten werden dabei in kleine Sektoren aufgeteilt, die dann den einzelnen Partitionen zugeteilt werden. Eine Partition kann aus Sektoren unterschiedlicher Bereiche der Festplatten bestehen.

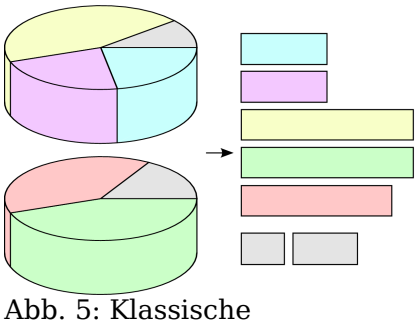
Abb. 5: Klassische Partitionierung

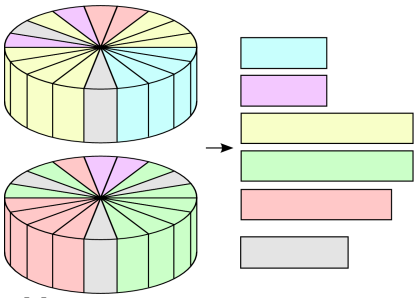
Abb. 4: Partitionierung mit LVM

Auch der Austausch von Festplatten gestaltet sich sehr einfach, da die Daten im laufenden Betrieb zwischen verschiedenen physikalischen Datenträgern hin- und hergeschoben werden können. Nur für den Umbau der Festplatten muss der Rechner abgeschalten werden.

Im Anhang findet sich eine Liste der wichtigsten LVM-Kommandos.

Verbund von RAID und LVM

Leider hat das LVM unter Linux keine oder nur sehr eingeschränkte RAID-Funktionalität. Dies kann allerdings dadurch umgangen werden, dass als physikalische Geräte bereits RAID-Verbünde verwendet werden. Damit ist die Datensicherheit bei Festplattenausfall gegeben, aber auch die Flexibilität eines LVM.

Tar

„tar“ ist ein Packprogramm, das aus mehreren Dateien oder Verzeichnissen eine große Datei macht. Ursprünglich war „tar“ dafür gedacht um ein Archiv auf ein Band zu kopieren, darum auch der Name „Tape Archiver“. Um Platz zu sparen kann man zusätzlich noch ein Komprimierungsprogramm verwenden, z.B. Gzip oder Bzip2. Tar wird in der aktuellen Version des Backups nur noch verwendet um einen Verzeichnisbaum vom Webserver herunterzuladen.

Rsync

Rsync ist ein mächtiges Programm zur Datenübertragung, entweder lokal zwischen verschiedenen Verzeichnissen oder aber auch zwischen verschiedenen Rechnern. Dabei wird nicht immer der gesamten Verzeichnisbaum übertragen, sondern es werden nur Dateien übertragen, die im Ziel noch nicht existieren bzw. die sich verändert haben. Es wird auch darauf geachtet dass die originale Struktur beibehalten wird, dass inkludiert sowohl Rechte der Dateien als auch symbolische und harte Verweise. Dies eignet sich damit ausgezeichnet um Backups durchzuführen. Mit verschiedenen Optionen kann die Arbeitsweise von Rsync beeinflusst werden.

Webpage

Um Statistiken zu den Backups zu sehen und um Daten aus dem Backup zu restaurieren gibt es eine dynamische Webseite auf dem Backupserver. Als Technologien kommen folgende zum Einsatz:

- Apache

  Als Webserver kommt der beliebte Webserver 'Apache' zum Einsatz. Es sollte aber mit allen PHP-fähigen Webservern funktionieren.

  Die Webseite selber ist mit HTAUTH geschützt, einer Methode um auf eine Webseite mit besonderen Rechten zuzugreifen.

- HTML/CSS

  HTML ist ein Akronym für „Hypertext Markup Language“ und stellt die Basis des Erfolgs des WWW dar. Es ist eine einfache Textformatierungssprache, die über die Erweiterung CSS (Cascading Style Sheets) mächtige Formatierungsregeln bekommen hat.

- PHP

  Als Programmiersprache kam PHP zum Einsatz, da es für Webseiten besser als Perl geeignet ist. Diese Web-Programmiersprache produziert dynamisch HTML um interaktive Webseiten zu ermöglichen. Ein Vorteil von PHP ist, dass es im Quelltext Bereiche geben kann, die reines HTML enthalten und andere in denen dann PHP-Code steht.

  Auch sonst ist PHP eine sehr mächtige Sprache, die mit Plugins erweiterbar ist. So kommt auch das Plugin GD zum Einsatz, dass eine mächtige Grafik-Bibliothek bietet. So können skriptmässig Grafiken und Bilder erzeugt werden, was im Falle des CGA-Backups für die Statistikauswertungen verwendet wird.

CGA-Backup

Clientseite

Als Client definiert sich die Maschine, für die das Backup gemacht wird. Auf diesem ist das wichtigste das Skript „cgabackup“, das vom Client gestartet wird. Bei den Servermaschinen geschieht das durch „Cron“. Auf anderen Rechnern, speziell auf Notebooks, wird das Skript händisch gestartet.

Das Skript „cgabackup“ ist das Herzstück des Backupsystems. Es führt aufgrund seiner Konfigurationsdateien das Backup durch.

```
cgabackup [-c configfile] [-v] [-p]
-c        Ort der zentralen Konfigurationsdatei
-v        Verbose; Gibt Informationen, was gerade getan wird
-p        Pausiert vor Beendigung (nur interaktiv sinnvoll)
```

Aus der zentralen Konfigurationsdatei (standardmäßig /etc/cgabackup.conf) werden globale Optionen und Beschreibungen für die einzelnen durchzuführenden Backups gelesen. Ein Beispiel könnte so aussehen:

```
host=backup.example.com
rsh=ssh

# Backup der Homeverzeichnisse
dir=/home/*
hostdir=/backup/myhost/home/

# Backup der Konfiguration
dir=/etc/
hostdir=/backup/myhost/etc/
```

Die verschiedenen Teile der Konfigurationsdatei sind jeweils durch Leerzeilen getrennt. Der erste Teil stellt die globale Konfiguration dar, die anderen Teile dann einzelne Backups, die der Reihe nach abgearbeitet werden. Kommentare werden mit dem Zeichen '#' eingeleitet.

Im Beispiel würde für jedes Verzeichnis in /home ein Backup gemacht werden, wobei bei der Verwendung von Wildcards dies noch an den Hostdir angehängt wird, also würde aus dem Verzeichnis 'bob' dann /backup/myhost/home/bob.

In den Verzeichnissen die dann gebackupt werden, kann auch noch eine gesonderte Konfigurationsdatei abgelegt werden, genannt „.cgabackup“. Der Punkt am Anfang des Namens markiert unter UNIX-Betriebssystemen versteckte Dateien, d.h. die Datei wird beim normalen Dateilisting nicht angezeigt. In dieser Datei können noch spezifische Optionen gesetzt werden:

```
BACKUP_MAILTO      bob@example.com
BACKUP_SUCCESSMAIL    1
BACKUP_ERRORMAIL      1
BACKUP_EXCLUDE        ./tmp/ *.mp3
```

- BACKUP_MAILTO

  Wenn das Backupsystem Mails verschickt, werden diese normalerweise an Verzeichnisname@hostname geschickt. Hier kann aber eine E-Mail-Adresse festgelegt werden.

- BACKUP_SUCCESSMAIL

  Immer wenn ein Backup durchgeführt wurde, wird ein Mail mit der Logdatei verschickt (impliziert BACKUP_ERRORMAIL) (Standard: nein).

- BACKUP_ERRORMAIL

  Nur im Falle, dass das Backup fehlgeschlagen ist wird ein Mail verschickt. (Standard: nein).

- BACKUP_EXCLUDE

  Hier wird eine Liste von Verzeichnissen und Datei-Wildcards angegeben, die nicht gebackupt werden.

Die Backups werden am Server im Verzeichnis 'hostdir'/YYYYMMDD abgelegt, so dass das Datum des Backups nachvollziehbar ist. Während des Backups wird das Verzeichnis 'hostdir'/YYYYMMDD_incomplete genannt, um darzustellen, dass das Backup noch nicht abgeschlossen ist. Daran ist auch erkennbar, ob ein Backup möglicherweise abgebrochen wurde.

Die Daten können von RSYNC auf verschiedene Weise übertragen werden. Entweder über RSH (Remote Shell, Standard), SSH (Secure Shell) oder über ein eigenes Protokoll. Das eigene RSYNC-Protokoll wird von CGA-Backup derzeit nicht unterstützt, eine Erweiterung ist aber geplant.

Da dieses Skript das Herzstück des Backup-Systems darstellt, ist der Quellcode im Anhang abgedruckt.

Serverseite

Serverseitig laufen verschiedene Skripte um die Backups vor- und nachzubereiten. Die automatisch gestarteten Skripts werden (wie bei den Servermaschinen) von „Cron“ gestartet. In /etc/cgabackup-server.conf befindet sich eine Konfigurationsdatei, die folgendes Aussehen hat:

```
root_dir=/backup
mail=admin@example.com
```

- 'root_dir'

  'root_dir' bezeichnet dabei ein Verzeichnis, von dem aus Backups gesucht werden.

- 'mail'

  Die Variable 'mail' enthält eine E-Mail-Adresse, an die unter anderem die Statistikauswertungen geschickt werden.

Wenn der Backupserver für verschiedene Hosts mit unterschiedlicher Konfiguration zuständig ist, können die Einträge vervielfacht werden, wobei diese durch Leerzeile getrennt werden. Natürlich sind auch Kommentare mit # am Zeilenanfang möglich.

Vor- und Nachbereiten eines Backups

Vom clientseitigen CGA-Backup werden vor und nach dem Backup Skripte aufgerufen (cga_pre_backup und cga_post_backup), die die folgenden Aufgaben erledigen:

- Check ob bereits ein Backup durchgeführt wird. Dafür wird eine Lock-Datei angelegt, die die PID des Parentprozesses enthält und erst bei Beendigung wieder gelöscht wird. Sollte bereits ein Backup aktiv sein (Lock-Datei existiert und Prozess mit dieser PID läuft), wird das neue Backup abgebrochen.
- Wenn bereits Backups an diesem Tag gemacht wurden, werden die alten Backups umbenannt, so dass aus YYYYMMDD YYYYMMDDa, aus YYYYMMDDa YYYYMMDDb gemacht wird u.s.w. Die betrifft natürlich auch die Logdateien und die unvollständigen Backups.

- Der wichtigste Schritt ist das Kopieren des letzten aktuellen Backups (das ist das, auf das last_backup zeigt. Sollte dieser Link nicht existieren, wird das letzte vollständige Backup genommen). Es wird also eine Kopie mit harten Verweisen des letzten Backups angelegt, damit in dieses vom CGA-Backup die Updates eingespielt werden können. Genannt wird das Verzeichnis YYYYMMDD_incomplete.
- Sobald das Backup abgeschlossen ist, wird das Verzeichnis auf den richtigen Namen umbenannt (also ohne _incomplete) und der symbolische Link last_backup auf dieses Verzeichnis umgebogen.

Löschen alter Backups

```
cga_del_backups [-c configfile]
-c        Gibt Ort der Konfigurationsdatei an
```

Jeden Tag, bevor die neuen Backups eingeteilt sind, läuft das Skript 'cga_del_backups', dass nach einer gewissen Strategie alte Backups löscht. Über ein serverseitiges Configfile können die Zeiträume in der die Backups aufgehoben werden verzeichnisweise konfiguriert werden. Dieses Configfile nennt sich backup.cfg und gilt jeweils für alle Unterverzeichnisse dieses Verzeichnisses. Dies ist die Syntax:

```
daily=14
incomplete=7
weekly=5
monthly=6
quarterly=5
```

Alle Backups werden prinzipiell 'daily' Tage aufgehoben, unvollständige Backups allerdings nur 'incomplete' Tage. Das jeweils erste Backup einer Woche wird für 'weekly' Wochen aufgehoben, das jeweils erste Backup eines Monats 'monthly' Monate und das jeweils erste Backup eines Quartals für 'quarterly' Quartale. Das neueste Backup bleibt auf jeden Fall bestehen, auch wenn es sich um ein unvollständiges handelt.

Statistiken berechnen

```
cga_build_statistic [-c configfile]
-c        Gibt Ort der Konfigurationsdatei an
```

Jeden Tag nachdem das Gros der Backups abgeschlossen ist, werden vom Skript 'cga_build_statistic' für jedes Verzeichnis Statistiken berechnet:

- Wie groß die Instanzen eines Backups absolut sind.
- Wie viel Platz die Instanzen eines Backups auf der Festplatte brauchen (Daten die zum nächstneueren Backup gleich geblieben sind brauchen keinen Platz).
- Einen Verlauf der Größe der jeweils aktuellsten Instanz eines Backups.
- Einen Verlauf der Gesamtgröße aller Backups eines Verzeichnisses.
- Für übergeordnete Backupverzeichnisse wird jeweils die Sume der Backups berechnet (z.B. für /backup/myhost die Summe von /backup/myhost/etc und /backup/myhost/home).

Diese Informationen werden beim Backup gespeichert um auf der Homepage dargestellt zu werden. Ausserdem wird per Mail eine Statistik ausgeschickt, in der aufgeführt wird, welche Backups (lezte Instanz bzw. Gesamt) am meisten gewachsen sind. Weiters enthält das Mail eine Liste aller Backups, in denen in den letzten drei Tagen kein neues vollständiges Backup erstellt wurde.

Verzeichnisinhalt

Ein typisches Backupverzeichnis sieht folgendermaßen aus:

```
20080401                          Dies sind Verzeichnisse die einzelne
20080701                          Instanzen des Backups beinhalten
20080719
20080722
20080723
20080724
cgabackup-20080401.log            Dies sind Logdateien zu den
cgabackup-20080701.log            jeweiligen Instanzen
cgabackup-20080719.log
cgabackup-20080722.log
cgabackup-20080723.log
cgabackup-20080724.log
last_backup                       Ein Link zur aktuellsten Instanz
statistic                         Statistikauswertung der Instanzen

statistic.last_backup.progress    Verlauf der Größe des akt. Backups
statistic.total.progress          Verlauf der Größe des gesamten Backups
```

Webpage

Wie bereits früher in diesem Dokument erwähnt gibt es eine Webpage um auf die Backups zuzugreifen. Dafür ist auf dem Backup-Server ein Apache mit PHP installiert.

Die Webpage soll vor allem folgende Ansprüche befriedigen:

- Zugriff auf die Backups um einzelne Dateien und Verzeichnisse zu rekonstruieren.
- Kontrolle ob die Backups erfolgreich durchgeführt wurden.
- Statistiken über Platzverbrauch abfragen.

Rechtesystem

Für die Authentisierung über HTAUTH befindet sich im Hauptverzeichnis der Webpage eine Datei namens .htaccess, in der geregelt ist, wer mit welchem Passwort darauf zugreifen kann. Üblicherweise geschieht das über ein Kopie der 'passwd'-Datei des Hauptservers.

Im Verzeichnis der Webpage befindet sich eine Datei 'conf', in der geregelt ist, wer worauf zugreifen darf. Diese Datei hat folgendes Format:

```
=Personal Backups
:/backup/myhost/home/%u:Your Homedirectory

=Other Backups
bob,@admin:/backup/myhost/www:Webserver
@sekr:/backup/myhost/sekr:Secretary stuff

=Admin Stuff
@admin:/backup/:All Backups
```

Es handelt sich also um jeweils einen Eintrag pro Zeile. Die Felder eines Eintrags werden mit ':' getrennt. Das erste Feld beschreibt wer darauf zugreifen darf. Ein leeres Feld bedeutet, dass alle User darauf zugreifen dürfen. Ein Name bestimmt einen bestimmten User. Ein Name der mit @ beginnt bezeichnet, dass User die zu einer Gruppe gehören dort zugreifen dürfen. Das zweite Feld eines Eintrags beschreibt den Pfad, wobei alle untergeordneten Verzeichnisse jeweils eingeschlossen sind. Ein %u wird durch den Namen des/der eingeloggten BenutzerIn ersetzt. Das dritte Feld enthält eine Beschreibung des Verzeichnisse, die in der Auflistung angezeigt wird.

Die mit '=' beginnenden Zeilen sind Überschriften. Leerzeilen werden ignoriert.

Dateizugriff

Eine Besonderheit beim Dateizugriff ist zu erwähnen. Es werden für die Dateilistings von Backupverzeichnissen und für die Downloads nicht die PHP-internen Funktionen (opendir, fopen, ...) verwendet sondern externen Programme die per SUID mit Adminrechten ausgeführt werden um das Problem zu umgehen, dass der Webserver als nichtpriviligierter Prozess keine Leserechte in diese Verzeichnisse bietet. Diese externen Programme sind in C programmiert und auf die minimale Funktion beschränkt um etwaige Sicherheitsprobleme zu minimieren.

Installation

Installation als Client

Aus dem Server-Verzeichnis der Distribution müssen die Dateien cgabackup und cgabackup.conf herauskopiert werden, erstere in ein bin-Verzeichnis (z.B. /usr/local/bin), zweitere in ein Konfigurationsverzeichnis (z.B. /etc).

Natürlich muß darauf geachtet werden, dass alle benötigten Programme installiert sind. Das sind Perl, Cron, rsync und rsh oder ssh. Damit der Rechner auch dazu in der Lage ist Mails zu verschicken, sollte ein Programm installiert sein, dass /usr/lib/sendmail zur Verfügung stellt (z.B. Sendmail, Exim oder Postfix).

Über einen Cron-Eintrag wird der Start des Programms eingeteilt, z.B.:

```
0 2 * * *       root    /usr/local/bin/cgabackup
```

Damit wird der Start auf jeden Tag um 2:00 festgelegt.

Der Zugriff über RSH oder SSH auf den Backupserver muss passwortlos möglich sein. Bei RSH muss dafür in der Datei .rhosts im Homeverzeichnis von root am Backupserver ein Eintrag für die Clientmaschine erstellt werden (Ein Eintrag pro Zeile: „hostname username“). Für SSH muss am Client mit Hilfe von 'ssh-keygen' eine Identifikation erstellt werden und der öffentliche Schlüssel (~/.ssh/id_rsa.pub) dann am Server der Datei /root/.ssh/authorized_keys hinzugefügt werden.

Installation als Server

Am Server werden die Skripte benötigt, die im server-Verzeichnis der Distribution liegen.

Das cgabackup-server.conf muß nach /etc kopiert und angepasst werden. Es kann natürlich auch der Ausführung der Skripte mit -c eine alternative Konfigurationsdatei angegeben werden.

Auch am Server müssen die beim Client angegebenen Programme installiert sein. Außerdem muß dort rsh-server oder sshd installiert sein.

Über Cron-Einträge werden Termine der zwei Wartungsskripte geplant:

```
5 0 * * *       root    /usr/local/bin/cga_del_backups
0 7 * * *       root    /usr/local/bin/cga_build_statistic
```

Zur Einrichtung des Logical Volume Managers gibt es im Anhang Informationen.

Installation der Webpage

Wenn auch noch die Webpage installiert werden soll, ist außerdem Apache und PHP mit libgd notwendig. Zum kompilieren der suid-Programme muss auch gcc und make installiert sein, dass kann aber auch auf einem anderen Rechner mit gleichem System passieren.

Die Dateien aus dem www-Verzeichnis in ein vom Webserver erreichbares Verzeichnis kopieren. Dort 'make' aufrufen, damit werden die benötigten Suid-Programme kompiliert und die Rechte umgesetzt.

Zur Konfiguration muß noch die 'conf'-Datei angepasst und per .htaccess das Verzeichnis vor unerlaubten Zugriffen geschützt werden. Die Usernamen die zur Authentisierung verwendet werden sind die gleichen, die dann auch für die 'conf'-Datei verwendet werden.

Konfiguration an der Abteilung für Computergraphik

Für diese Arbeit zeige ich die Implementation des CGA-Backups am Beispiel der Abteilung für Computergraphik der TU-Wien, die dort den Hauptteil der Backupstrategie darstellt. Der Backupserver ist räumlich getrennt bei einem anderen Institut aufgestellt, um auch im Katastrophenfall eine Sicherung zu haben. Erreichbar ist der Server als backupcga.iemar.tuwien.ac.at .

Der Backupserver bindet mehrere Raid-Systeme zu einer LVM-Gruppe zusammen, in denen für die verschiedenen Hosts eigene logische Laufwerke angelegt wurden, um zu verhindern, dass ein zu volles Backup gleich alle anderen Backups behindert. Als Verzeichnisname dient jeweils /backup-cg/hostname/backupname, z.B. /backup-cg/erzherzog/etc oder /backup-cg/zwirn/homes/wp.

Alle Hosts liegen auf einem eigenen logischen Laufwerk, die Homeverzeichnisse der User haben auch jeweils ein eigenes logisches Laufwerk, da sich diese sehr oft spontan um große Datenmengen ändern, womit ein Konflikt mit anderen Verzeichnissen verhindert werden soll.

Hier ein Auszug aus dem 'df -h' (Anzeige des Füllungsgrades der Laufwerke):

```
/dev/mapper/backup--cg-default
                      9.9G  151M  9.2G   2% /backup-cg
/dev/mapper/backup--cg-christl
                      9.9G  5.7G  3.8G  61% /backup-cg/christl
/dev/mapper/backup--cg-erzherzog
                      168G  139G   21G  88% /backup-cg/erzherzog
/dev/mapper/backup--cg-erzherzog--homes
                       99G   88G  6.3G  94% /backup-cg/erzherzog/homepages
/dev/mapper/backup--cg-zwirn
                       99G   19G   75G  21% /backup-cg/zwirn
/dev/mapper/backup--cg-zwirn--homes
                      640G  589G   19G  97% /backup-cg/zwirn/homes
```

Damit nicht alle Rechner eine Kopie des cgabackup-Skriptes benötigen, ist dieses in /usr/cg/bin zu finden, welches auf allen Rechnern gemountet wird. Bevorzugt sollten alles Backups zwischen 0:30 und 7:00 laufen, da vorher die alten Backups gelöscht werden und nachher die Statistiken erstellt werden.

CGA-Backup für MS Windows-Computer

Damit das CGA-Backup auch unter Microsoft Windows läuft wird das Programm 'Cygwin' benötigt. Cygwin ist eine GNU-Distribution für Windows und bietet damit die üblichen Programme, unter anderem auch Rsync, Perl und SSH.

Da Cygwin keine Laufwerke wie Windows kennt, sondern einen einheitlichen Namensraum wie UNIX verwendet, ist die Pfaddeklaration eine andere. So ist das Laufwerk C: als /cygdrive/c erreichbar.

Damit nicht jeder Windows-Computer eine eigene Cygwin-Installation benötigt, gibt es am Zwirn eine Cygwin-Installation (\\zwirn\cygwin), die über Samba verwendet werden kann. Dazu muß die Registry angepasst werden, die entsprechenden Keys finden sich im Verzeichnis DOC (dort gibt es auch ein HOWTO). Im Hauptverzeichnis sind zwei Skripte wesentlich: start_cygwin.bat, dass eine Shell auf dem Rechner öffnet und dobackup.bat, welches ein Backup startet, wobei die Konfigurationsdatei in c:\cgabackup.conf erwartet wird.

Anhang

Die wichtigsten Kommandos für LVM

```
# pvdisplay
```
Listet alle physikalischen Geräte auf.

```
# vgdisplay
```
Listet alle verfügbaren Laufwerkgruppen auf.

```
# lvdisplay -m
```
Listet alle angelegten Laufwerke auf. Der Parameter -m bewirkt, dass auch angezeigt wird auf welchen phys. Geräten das Laufwerk liegt.

```
# pvs
# vgs
# lvs
```
Kurzübersichten über phys. Geräte (pv), Laufwerksgruppen (vg) und logische Laufwerke (lv).

```
# pvcreate /dev/mdX
```
Device 'mdX' als phys. Gerät aktivieren.

```
# vgextend backup /dev/mdX
```
Phys. Gerät /dev/mdX der Laufwerkgruppe 'backup' hinzufügen.

```
# lvcreate -n name -L 4G backup
# mkfs.ext3 /dev/backup/name
```
In der Laufwerkgruppe 'backup' wird ein Laufwerk mit der Größe 4 GB namens 'name' angelegt. Das Device ist dann /dev/backup/name. Das zweite Kommando legt darauf ein EXT3-Dateisystem an.

```
# lvextend -L 8G /dev/backup/name
# ext2resize /dev/backup/name
```
Das Laufwerk 'name' in der Laufwerkgruppe 'backup' wird auf 8 GB vergrößert und das Filesystem danach auf die neue Größe angepasst.

```
# lvremove /dev/backup/name
```
Das Laufwerk 'name' in der Laufwerkgruppe 'backup' wird entfernt.

```
# pvmove /dev/mdX
# vgreduce backup /dev/mdX
# pvremove /dev/mdX
```
Diese Kommandos sind notwendig um ein phys. Gerät aus dem LVM zu entfernen. Mit dem ersten Kommando werden alle Laufwerke auf andere physikalische Geräte verlagert. Natürlich muss entsprechend viel Platz vorhanden sein und das nimmt auch einige Zeit in Anspruch. Das zweite Kommando entfernt das phys. Gerät dann aus der Laufwerkgruppe, das letzte generell aus dem LVM.

Literaturverzeichnis

Easy Automated Snapshot-Style Backups with Linux and Rsync:
http://www.mikerubel.org/computers/rsync_snapshots/ (2004-01-04)

Software RAID HOWTO:
http://www.tldp.org/HOWTO/Software-RAID-HOWTO.html (v1.1, 2004-06-03)

LVM HOWTO:
http://www.tldp.org/HOWTO/LVM-HOWTO/ (0.19, 2006-11-27)

PHP-Dokumentation:
http://www.php.net/manual/en/ (5.2.6, 2008-08-01)

Apache:
http://httpd.apache.org/ (2.2.6, 2008-06-13)

Perl:
http://perldoc.perl.org/ (5.10.0, 2007-12-18)

Rsync:
http://samba.anu.edu.au/rsync/documentation.html (3.0.3, 2008-06-29)

tar:
http://www.gnu.org/software/tar/ (1.20, 2008-04-14)

EXT3:
http://e2fsprogs.sourceforge.net/ (1.41.0, 2008-07-10)

ext2resize:
http://ext2resize.sourceforge.net/ (1.1.19, 2004-09-30)