

# Rsyncbackup

Letzte Aktualisierung Montag, 2. Oktober 2006

Backup ist zwar lästig, aber wenn es im Ernstfall fehlt oder schlecht erreichbar ist, hat man ganz andere Probleme. Die Aufgabenstellung ist also, ein Backup zu bauen, welches schnell durchläuft, schnelles restore erlaubt und möglichst transparent und einfach funktioniert, damit es auf vielen Systemen eingesetzt werden kann.

## Die Lösung

Die Lösung heißt rsync. rsync ist ein kleines Programm, welches die Aufgabe, Daten von einem Platz zum anderen in maximaler Geschwindigkeit zu transportieren, hervorragend erfüllt. Gerade der leistungsschonende Transport von großen Dateien, die sich seit dem letzten Backup nur an wenigen Stellen verändert haben, ist optimal gelöst: Er werden nur die veränderten Daten übertragen, die anderen sind ja schon da.

Diese Lösung bedient sich eines schnellen Backuprechners mit viel Plattenplatz. Mindestens das Doppelte an Plattenplatz, wie die zu sichernden Server an Daten beinhalten, sollte vorhanden sein. Es empfiehlt sich aus Kostengründen ein IDE-Raid-System. Dieses System darf nicht das einzige vorhandene Backup sein, da es natürlich selbst wegen der Festplatten recht anfällig ist. Es muß also entweder dieses System selbst oder alle anderen Systeme nochmal auf ein weiteres System gesichert werden, welches höhere Sicherheitsansprüche erfüllt.

## Screenshots

## Details

Dieses Programm erfüllt die folgenden Anforderungen:

- Einfachst zu installieren
- Einfachstes Einbinden eines neuen zu sichernden Servers
- Schnelles restore
- Ergebnisse der Sicherungen auf einen Blick
- Datentransfer bei Sicherung per SSH verschlüsselt
- Möglichst wenig Installation/Voraussetzungen beim zu sichernden Client
- Vorhalten mehrerer Generationen vollständiger Backups bei minimalem Platzbedarf (Der Platzbedarf ist gleich dem des zu sichernden Servers plus aller im Generationszeitraum dort veränderter Dateien)

Dieses Programm ist allerdings limitiert insofern, als dass es (noch) kein komfortables restore erlaubt (das kommt aber noch) und dass es nur auf tägliche Backups ausgelegt ist. Backups mehrfach täglich sind zwar im Prinzip möglich, aber bringen die Logs durcheinander - das war aber auch nicht die Aufgabenstellung. Die Generationssicherungen sind linear, es ist also (noch) nicht möglich, z.B. ein Backup des Monatsersten aufzuheben. Es werden immer nur die letzten x Backups aufbewahrt.

Das Backupprogramm besteht aus einem Shellsript, welches von weiteren Shellsripten aufgerufen wird. Das Analyseprogramm ist ein simples Perlscript. Ein gewöhnliches Linuxsystem out-of-the-box genügt den Ansprüchen, wenn das Modul `Date::Calc` nachinstalliert worden ist und `rsync` vorhanden ist.

Der Backupserver sollte mit sehr schnellen Platten ausgestattet sein und viel Plattenplatz. Zur Zeit betreiben wir ein AMD-Doppelprozessorsystem mit 4,5 TB Hardware-IDE-Raid.

Für den Zugriff auf die Statistiken, Logdaten und das restore ist ein Apache erforderlich. Dieser sollte unbedingt mit SSL-Unterstützung installiert und verwendet werden.

- `Date::Calc` zu Datumsberechnungen.
- (`Data::Dumper` nur für Debugzwecke)

## Installation

Vorausgesetzt wird ein lauffähiges System mit Perl und Webserver. Der Pfad für das Backup muss auf derselben Partition liegen wie der Pfad, der die Generationen aufnehmen soll. Der Grund ist, daß aus Platzersparnisgründen Hardlinks verwendet werden. In den Scripten wird `/export/backup` und `/export/generation` verwendet.

- Installation der Perlmodule aus obiger Liste
- Kopieren des Programmes in ein beliebiges Verzeichnis des Backupservers. In den Scripten ist `/root/backup` konfiguriert.
- Erstellen eines Schlüsselpaares per `ssh_keygen`. Es sollte ein separater Schlüssel (`-f`) erstellt werden, der z.B. `rsync-key` heisst. Der Schlüssel darf keine Passphrase haben!
- Das Script `analysebackuplogs.pl` wird in die Crontab eingetragen, wertet die Logs aus und erzeugt eine Webseite, die die Ergebnisse der Sicherungen übersichtlich darstellt. Die Ausgabe des Scriptes muss in den Namen der gewünschten html-Datei umgeleitet werden.  
Die folgenden Aufgaben sind für jeden einzelnen Server, der zu sichern ist, vorzunehmen.
- Kopieren des public Key des oben erzeugten Schlüsselpaares auf den zu sichernden Server. Falls dort nicht vorhanden, eine `/root/authorized_keys` anlegen und per `chmod` auf 600 setzen. In diese Datei den öffentlichen Schlüssel einfügen.
- In der Configdatei des `sshd` auf dem zu sichernden Server muss `PermitRootLogin` auf `Yes` stehen (Default).
- Jetzt muss ein testweises `ssh remote-server` als `root` auf dem Backupserver ohne Passwortabfrage eine Session auf dem remote-server öffnen.
- `rsync` auf dem zu sichernden Server installieren
- Auf dem Backupserver eine `backup-(remoteservername).sh` nach vorliegendem Beispiel erzeugen und den Servernamen eintragen.
- Das Aufrufen dieses Shellsriptes sollte direkt zur ersten Sicherung führen.
- Wenn das geklappt hat, einen entsprechenden Cronjob erzeugen.  
Das restore muss bis zur endgültigen Version, die das im Webbrowser erlaubt, zu Fuß per `rsync` erfolgen.  
Historie

- 1.2 10/05: Befehle direkt vor und nach dem Backup auf dem remote Server ausführen möglich

- 1.1 10/05: Erste Version veröffentlicht

Download

Dieses Programm ist kostenfrei im nichtkommerziellen Umfeld verwendbar. Das Programm darf ohne meine Einwilligung nicht kommerziell verwendet werden. Die Stellen, an denen mein Name steht, dürfen nicht verändert werden; die Stellen, an denen mein Name in den html-Ausgaben auftaucht, dürfen ebenfalls nicht verändert oder gelöscht werden.

Selbstverständlich übernehme ich keine Haftung jeglicher Art für irgendwelche Schäden, die direkt oder indirekt durch die Nutzung dieses Programmes entstehen könnten.

Fehlerberichte und Anregungen zur Weiterentwicklung sind jederzeit willkommen.

Ich freue mich auch über eine Benachrichtigungsmail, wenn das Programm irgendwo eingesetzt wird.

- Das Backupsript
- Dasselbe, lesbar im Browser
- Der Wrapper, der für jeden Server einmal angelegt wird
- Dasselbe, lesbar im Browser
- Perlscript zum Analysieren der Logs
- Dasselbe, lesbar im Browser