

Rsync: Das beste Backupsystem



by Brian Hone
<bhone(at)eink.com>



Abstract:

About the author:

Brian Hone ist Systemadministrator und Softwareentwickler bei E-Ink Corp. In seiner Freizeit surft er in eiskaltem Wasser oder er hängt an Felswänden.

Datensicherung ist eine der schwierigsten und vernachlässigsten Aufgaben der Systemadministration. Es ist auch eine der wichtigsten Massnahmen. Es ist der letzte Schutz gegen Hardwareausfall, Sicherheitsverletzungen und die grösste aller Gefahren: die Benutzer. Obwohl es viele Backupsysteme gibt, die Tausende von Dollar kosten und auf teuren Bandlaufwerken mittels fehlerhafter proprietärer Software archivieren, gibt es eine bessere Lösung: Rsync und ein preiswertes Festplatten-Array.

Das Problem

Ich könnte hier eine lange Liste der Gründe aufführen, warum Datensicherung der Alptraum des Systemadministrators ist – falls Sie ein Administrator sind, ist das aber wahrscheinlich überflüssig. Einige der Gründe sind: teure Hardware (die meistens nicht funktioniert), teure Software – ein Alptraum für das Management – und viele verschwendete Stunden, um Mehrfachversionen von Dateien zu restaurieren. Der Höhepunkt ist die meistens fehlende Priorität der Geschäftsleitung für Sicherheitskopien, bis zum unausweichlichen Moment, wenn diese dringend gebraucht werden. Falls Sie Datensicherung/Wiederherstellungen durchgeführt haben, hatten Sie bestimmt ein ähnliches Gespräch:

User: "Ich habe eine Datei verloren, ich brauche die unbedingt zurück." "

SysAdmin: "Ok, was ist der Name?"

User: "Weiss ich nicht, ich glaube es war ein 'e' im Namen."

SysAdmin: "Ok, in welchem Verzeichnis war sie?"

User: "Keine Ahnung, es könnte eines von diesen dreien sein."

SysAdmin: "*Seufz*. Können Sie sich erinnern, wann Sie die Datei zuletzt gebraucht haben?"

User: "Ach...ich glaube, es war an einem Donnerstag im Februar oder im April. Was ist das Problem? Ich dachte ihr Leute habt eine Datensicherung für diese Art von Problem."

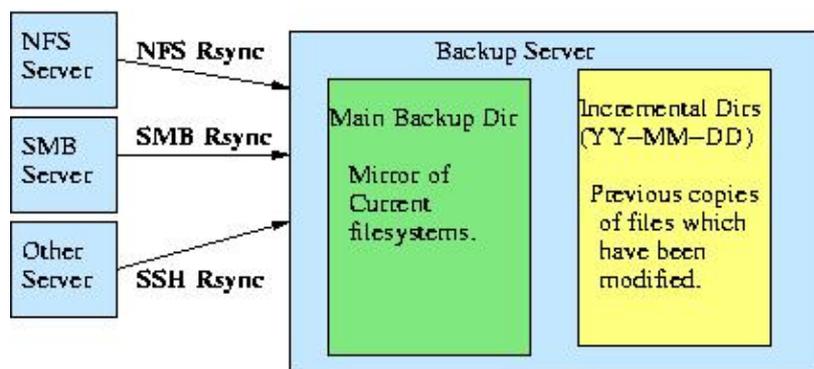
Die Rsync-Lösung

Rsync ist eine sehr wirksame Implementation eines hübschen kleinen Algorithmus. Seine Hauptwirkung liegt in der Fähigkeit, ein Dateisystem wirkungsvoll zu spiegeln. Mittels Rsync ist es einfach, ein System einzurichten, das eine aktuelle Kopie eines Dateisystems mittels einer flexiblen Anordnung von Netzwerkprotokollen (wie nfs, smb oder ssh) erzeugt. Außerdem kann ein auf Rsync basierendes Datensicherungssystem, alte Kopien von geänderten oder entfernten Dateien. Rsync hat zu viele Eigenschaften, um sie alle in diesem Artikel aufzuführen. Ich empfehle wärmstens, an dieser Stelle nachzulesen rsync.samba.org.

Das System

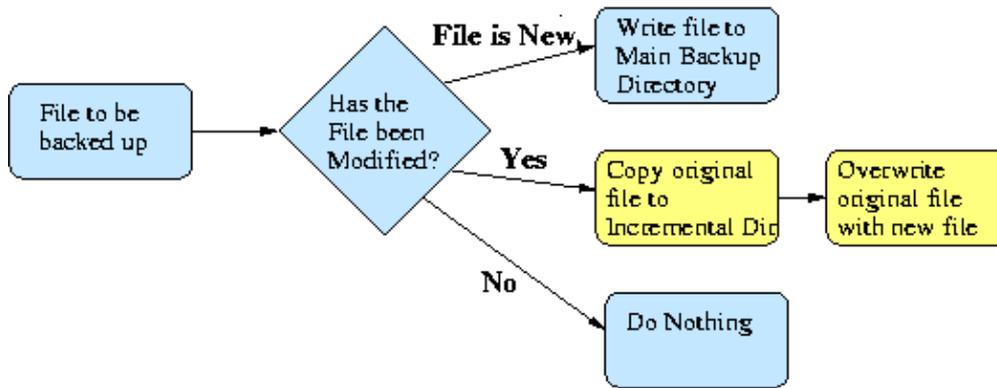
Kurz gesagt: unser System besteht aus einer billigen Linuxbox mit einigen preiswerten Festplatten und einem kleinen Shellsript, das rsync aufruft. **[Fig 1]** Wenn wir eine Datensicherung durchführen, befehlen wir rsync ein Verzeichnis namens 'YY-DD-MM' als Speicher für inkrementelle Änderungen einzurichten. Als nächstes durchsucht rsync die Server – von denen Sicherheitskopien hergestellt werden sollen – auf Dateiänderungen. Ist eine Datei geändert worden, wird die alte Version in das Inkrementellverzeichnis kopiert und dann die Datei im Hauptdatensicherungsverzeichnis überschrieben. **[Fig 2]**

Figure 1: Structure of an Rsync Backup Server



Im Allgemeinen beträgt das Volumen der Änderungen des Tages nur einen kleinen Prozentsatz des gesamten Dateisystems. Bei mir beträgt die typische durchschnittliche Grösse zwischen 0,5 % bis 1%. Mit einigen Backup-Festplatten mit der doppelten Speicherkapazität unserer Server können wir schrittweise Sicherungskopien für 50 – 100 Tage unterbringen. Wenn die Festplatten voll sind, tauschen wir sie mit neuen aus und lagern die alten an einem sicheren Ort. In der Praxis ist es möglich, inkrementelle Backups von über sechs Monaten auf Festplatten zu speichern. Falls Platz auf einem anderen Server vorhanden ist, können Sie die inkrementellen Backups dort speichern, bevor Sie die Festplatten wechseln, auf diese Weise können Sie die Anzahl der inkrementellen Backups beliebig groß ansetzen.

Figure 2: Backup of a file using Rsync



Die Vorteile: Einfache Wiederherstellung nach einem Totalverlust auch für einzelne Dateien

Zurück zu unserer imaginären Unterhaltung. Anstelle eines umständlichen Bandspeichersystems warten die inkrementellen Backups der letzten sechs Monate friedlich in unserer Linuxbox. Mit unser bevorzugten Kombination aus locate/find/grep finden wir alle Dateien unseres imaginären Users, die ein 'e' enthalten und welche die Zeitmarkierung eines Donnerstags im Februar oder im April enthalten. Diese Dateien laden wir in des Users Homeverzeichnis. Das Problem, herauszufinden, welche Version die richtige ist, ist damit zu meiner bevorzugten Art von Problemen geworden: das von einem anderen.

Als nächstes unser Lieblingsszenario – Totalverlust. Nehmen wir an, wir verlieren einen grösseren nfs/samba-Server. Von unserer gesicherten Sambakonfiguration können wir unseren Backupserver als Read-Only-Ersatz innerhalb von Minuten einsetzen. Probieren wir das mal mit dem Magnetband.

Wie sieht der Vergleich Rsync/Festplatten- Backup gegenüber Backup auf Magnetband aus

	Magnetband-Backup	Rsync
Kosten	Sehr hoch	Niedrig
Volle Datensicherung	Schnell	Schnell
Teilweise Datensicherung	Schnell	Schnell
Volles Wiederherstellen	Sehr langsam, evtl. von mehreren Bändern	Schnell – es ist alles auf der Festplatte
Datei wiederherstellen	Langsam, evtl. von mehreren Bändern, richtige Version oft schwer zu finden.	es ist alles auf der Festplatte und wir haben die volle Wirkung der UN*X Suchwerkzeuge wie, grep und locate
Totalverlust	Die einzige Option ist die vollständige Wiederherstellung	Kann schnellstens in einen Dateiserver umgestellt werden.

Die Werkzeuge

Es gibt viele Möglichkeiten, es aufzusetzen. Alle Werkzeuge sind Open-Source, sie sind in den meisten Distributionen enthalten und sie sind sehr flexibel. Nachfolgend beschreiben wir eine mögliche Einrichtung – es gibt aber noch viele andere Wege.

- **Der Server:** Ich benutze RedHat Linux. Es sollte mit jeder Distribution, jeder UN*X-Version funktionieren (Ich hab es sogar auf Mac OS X eingerichtet) Eine Einschränkung: viel RAM hilft.
- **Festplatte:** Nach meiner Erfahrung ist die kostengünstigste Lösung eine PCI-Firewirekarte, verbunden mit einer Reihe preiswerter Festplatten in externen Firewiregehäusen. Linux für eine grosse RAID-Partition einzurichten ist ziemlich schmerzlos.
- **Die Software:** Rsync ist ein grossartiges Werkzeug – es ist so eine Art Taschenmesser zum Spiegeln von Dateisystemen. Falls Sie es nicht kennen, gehen Sie zu rsync.samba.org !
- **Verbindung zu den Dateiservern:** Rsync ist sehr flexibel. Wir benutzen nfs und smbfs. Sie können auch durch Laufen lassen eines Rsyncdaemons auf einem Dateiserver Rsyncs eigenes Netzwerkprotokoll benutzen. Sie können Rsync auch anweisen, mittels ssh entfernte Sites zu sichern. Sehen Sie sich die Links unten an, um Informationen zum Aufsetzen solcher Verbindungen zu bekommen.

Das Script aufsetzen

Die Grundform dieses Skripts kam von der Rsync-Webseite. Er besteht eigentlich nur aus einem Befehl:

```
rsync --force --ignore-errors --delete --delete-excluded --exclude-from=exclude_file --backup --backup-dir=`date +%Y-%m-%d` -av
```

Die Hauptoptionen hier sind:

- `--backup`: Backups von Dateien erzeugen, bevor diese überschrieben werden
- `--backup-dir=`date +%Y-%m-%d``: Backupverzeichnis erzeugen für Backups, die folgendermassen aussehen: 2002-08-15
- `-av`: Archivmodus und Ausführlichmodus.

Das folgende Skript kann man jeden Abend mit cron unter Linux ausführen. Um das Skript jeden Abend um 11 Uhr zu starten, benutzen wir das Kommando "crontab -e" und dann geben wir das folgende ein:

```
0 23 * * * /path/to/your/script
```

Der Skript

Hier ist mein Shellskript, um das alles zusammenzufassen. Dabei gibt es wieder viele Möglichkeiten. Das folgende Script ist nur eine Implementation.

```
#!/bin/sh
#####
# Script to do incremental rsync backups
```

```

# Adapted from script found on the rsync.samba.org
# Brian Hone 3/24/2002
# This script is freely distributed under the GPL
#####

#####
# Configure These Options
#####

#####
# mail address for status updates
# - This is used to email you a status report
#####
MAILADDR=your_mail_address_here

#####
# HOSTNAME
# - This is also used for reporting
#####
HOSTNAME=your_hostname_here

#####
# directory to backup
# - This is the path to the directory you want to archive
#####
BACKUPDIR=directory_you_want_to_backup

#####
# excludes file - contains one wildcard pattern per line of files to exclude
# - This is a rsync exclude file. See the rsync man page and/or the
#   example_exclude_file
#####
EXCLUDES=example_exclude_file

#####
# root directory to for backup stuff
#####
ARCHIVEROOT=directory_to_backup_to

#####
# From here on out, you probably don't #
# want to change anything unless you #
# know what you're doing. #

#####
# directory which holds our current datastore
CURRENT=main

# directory which we save incremental changes to
INCREMENTDIR=`date +%Y-%m-%d`

# options to pass to rsync
OPTIONS="--force --ignore-errors --delete --delete-excluded \
--exclude-from=$EXCLUDES --backup --backup-dir=$ARCHIVEROOT/$INCREMENTDIR -av"

export PATH=$PATH:/bin:/usr/bin:/usr/local/bin

# make sure our backup tree exists
install -d $ARCHIVEROOT/$CURRENT

# our actual rsyncing function
do_rsync()
{
    rsync $OPTIONS $BACKUPDIR $ARCHIVEROOT/$CURRENT
}

```

```

}

# our post rsync accounting function
do_accounting()
{
    echo "Backup Accounting for Day $INCREMENTDIR on $HOSTNAME:">/tmp/rsync_script_tmpfile
    echo >> /tmp/rsync_script_tmpfile
    echo "#####">>/tmp/rsync_script_tmpfile
    du -s $ARCHIVEROOT/* >> /tmp/rsync_script_tmpfile
    echo "Mail $MAILADDR -s $HOSTNAME Backup Report < /tmp/rsync_script_tmpfile"
    Mail $MAILADDR -s $HOSTNAME Backup Report < /tmp/rsync_script_tmpfile
    echo "rm /tmp/rsync_script_tmpfile"
    rm /tmp/rsync_script_tmpfile
}

# some error handling and/or run our backup and accounting
if [ -f $EXCLUDES ]; then
    if [ -d $BACKUPDIR ]; then
        # now the actual transfer
        do_rsync && do_accounting
    else
        echo "cant find $BACKUPDIR"; exit
    fi
else
    echo "cant find $EXCLUDES"; exit
fi

```

Ressourcen

- Rsync: <http://rsync.samba.org/>
- NFS: <http://nfs.sourceforge.net/nfs-howto>
- SMBFS: <http://samba.org/>
- Linux RAID: <http://linas.org/linux/raid.html>

Webpages maintained by the LinuxFocus Editor team

© Brian Hone

"some rights reserved" see linuxfocus.org/license/

<http://www.LinuxFocus.org>

Translation information:

en --> -- : Brian Hone <bhone(at)eink.com>

en --> de: Jürgen Pohl <sept.sapins(at)verizon.net>