

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

Over de auteur:

Brian Hone is systeembeheerder en software-ontwikkelaar bij E Ink corporation. In zijn vrije tijd besurft hij koude waters en bungelt hij aan rotswanden.

Vertaald naar het Nederlands door:
Floris Lambrechts
<floris(at)linuxfocus.org>

Rsync: Het ultieme Back-up Systeem



Kort:

Back-ups maken is tegelijk één van de belangrijkste en één van de meest onderschatte onderdelen van het systeembeheer. Het is het laatste redmiddel in geval van hardwarefouten, inbraken en - de ergste bedreiging van allemaal - eindgebruikers. Er bestaan ontelbaar veel backup systemen die duizenden dollars kosten en archiveren op dure tape drives met niet-vrije, brakke software, maar er is een betere oplossing: Rsync en een array van goedkope schijven.

Het Probleem

Ik zou een waslijst aan redenen kunnen opsommen waarom back-ups de nachtmerrie zijn van de systeembeheerders. Als je er zelf één bent, dan hoeft ik je er waarschijnlijk niks meer over te vertellen. Een greep uit het aanbod: dure hardware die vaker wel dan niet stuk is, dure software die extreem moeilijk te beheren is, en lange nachtelijke uren om verschillende versies van bestanden te herstellen. Om de zaak nog erger te maken hechten bedrijven over het algemeen erg weinig belang aan het maken van back-ups; tot de dag aanbreekt waarop ze nodig zijn uiteraard. Als je een beetje ervaring hebt in de sector van backup & restore, heb je waarschijnlijk al wel conversaties gehad zoals deze:

Gebruiker: "Ik ben een bestand kwijt. Ik wil dat je het meteen terugzet."

SysAdmin: "Ok, wat is de naam van het bestand?"

Gebruiker: "Geen idee, ik denk iets met een 'e' erin."

SysAdmin: "Ok, in welke directory stond het?"

Gebruiker: "Dat weet ik niet, het zou kunnen in één van deze drie..."

SysAdmin: "*Zucht* Weet je wanneer je het bestand voor het laatste hebt gebruikt?"

Gebruiker: "Wel... Ik denk dat het een dinsdag was in februari of april. Wat is het probleem? Ik dacht dat jullie een backup systeem hadden om zulke dingen op te lossen?"

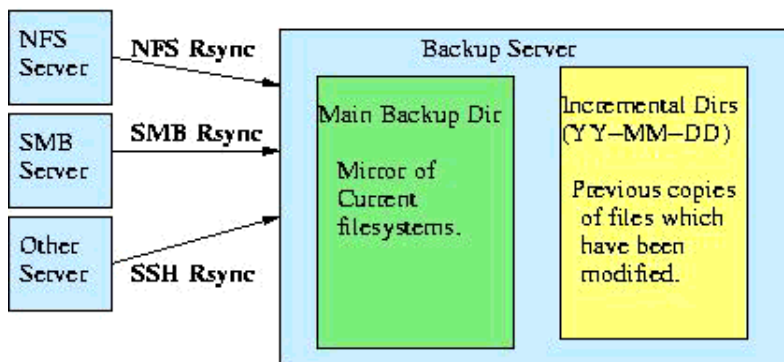
Het Rsync Alternatief

Rsync is een krachtige implementatie van klein elegant algoritme. Zijn grote kwaliteit is dat het toelaat om heel efficiënt een kopie van een bestandssysteem bij te houden. Met rsync kun je heel snel een systeem opzetten waarmee je steeds een up-to-date kopie hebt van een bestandssysteem, of dit nu beschikbaar is via nfs, smb of ssh. Een andere sterke kant van rsync is de mogelijkheid om ook oude versies van bestanden bij te houden - bestanden dus die gewijzigd of zelfs verwijderd werden. Rsync heeft nog veel meer mogelijkheden, we kunnen ze onmogelijk allemaal vermelden. Een bezoekje aan rsync.samba.org is zeker aan te bevelen.

Het Systeem

Om kort te gaan: we gebruiken een goedkope Linux doos met veel goedkope opslagruimte en een klein shell-scriptje om rsync aan te roepen. [Fig 1] Bij het maken van de back-up vertellen we rsync dat het een directory moet maken volgens het schema 'YY-DD-MM' - dit is de plaats waar de incrementele veranderingen zullen worden opgeslagen. Vervolgens onderzoekt rsync de gewenste servers op veranderingen. Als een file veranderd is, kopieert rsync de oude versie naar de incrementele directory en overschrijft het dit bestand in de primaire backup directory. [Fig 2]

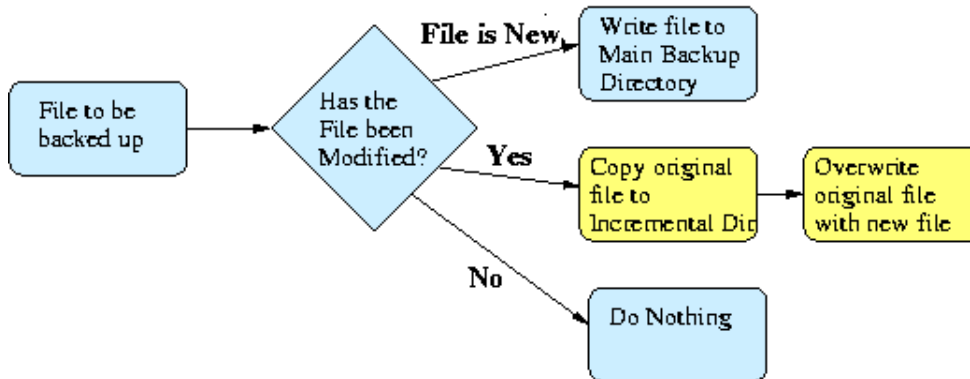
Figure 1: Structure of an Rsync Backup Server



In het algemeen zullen de veranderingen van 1 dag slechts een fractie uitmaken van de totale grootte van het bestandssysteem. In mijn ervaring gaat het om zo'n 0.5% tot 1%. Daarom kunnen we met een backup-volume van twee maal het volume van de file servers 50 tot 100 dagen incrementele backups op schijf bijhouden. Wanneer de schijven vol geraken verhuis je ze naar een veilige locatie (off-site) en vervang je ze door een nieuw stel schijven. In de praktijk is het mogelijk om meer dan zes maanden incrementele backups bij te houden. Als je er de ruimte voor vindt kun je de oude 'incrementals' zelfs kopiëren naar een andere server voordat je de schijven verwisselt. Op die manier kun je een onbeperkte

geschiedenis van incrementele backups op harde schijf beschikbaar houden.

Figure 2: Backup of a file using Rsync



De voordelen: eenvoudig disaster recovery en herstellen van bestanden

Denk even terug aan de imaginaire conversatie van hierboven. In de plaats van een onhandig systeem met magnetische banden heb je nu een half jaar backup-geschiedenis gewoon beschikbaar op je Linux systeem. Met je favoriete combinatie van locate/find/grep kun je alles terugvinden waar de imaginaire gebruiker ooit mee bezig geweest is, met inbegrip van die bestanden 'met een e erin' die gedateerd zijn op een dinsdag in februari of april. Vervolgens drop je ze in de home directory van de gebruiker in kwestie en klaar is kees. Het probleem van het uitzoeken welk bestand nu eigenlijk het juiste is, is omgetoverd in mijn favoriete categorie van problemen: die van iemand anders.

Nu gaan we ons favoriete scenario doorlopen - de complete crash. Stel, je hebt een grote nfs/samba server die stuk gaat. Wel, als je een kopie hebt van de samba configuratie dan kun je in enkele minuten een read-only vervanging online brengen. Dat wil ik je wel eens zien doen met een tape back-up.

Rsync back-up op schijf tegenover Tape back-up

	Tape Backup	Rsync
Kost	Zeer hoog	Laag
Volledige Back-up	Snel	Snel
Incrementele Back-up	Snel	Snel
Volledig terugzetten	Zeer traag, waarschijnlijk verspreid over meerdere tapes.	Snel - het staat op schijf.
Bestanden terugzetten	Traag, misschien op verschillende tapes, correcte versie vaak moeilijk te vinden.	Zeer Snel - het staat op schijf en je kunt gebruik maken van UN*X tools zoals find, grep en locate.
Volledige Crash	De enige optie is de server volledig terugzetten.	De backup server kan dienen als tijdelijke file-server.

De Tools

Er zijn verschillende manieren om een dergelijk systeem op te zetten. Al de tools zijn open source, zitten in alle distributies en zijn heel flexibel. We beschrijven hier één mogelijke setup, maar het is lang niet de enige mogelijke aanpak.

- **De Server:** Ik gebruik RedHat Linux. Elke distributie zou het moeten doen, net als eender welke UN*X. (Ik heb het zelfs gebruikt onder Mac OS X.) Één opmerking: een grote hoeveelheid RAM-geheugen kan helpen.
- **Harde Schijf:** De makkelijkste manier die ik gevonden heb om een groot volume te bouwen met goedkope schijven is een PCI FireWire kaart verbonden met een stel goedkope IDE schijven in externe FireWire behuizingen. Linux instellen om ze te gebruiken als 1 groot RAID volume gaat vrij pijnloos.
- **De Software:** Rsync is fantastisch. Het is een soort Zwitsers officiersmes voor 'file system mirroring'. Als je het nog niet kent, bekijk het dan op rsync.samba.org.
- **Verbinding met de File servers:** Rsync is heel flexibel. Wij gebruiken nfs en smbfs. Je kunt ook rsync's eigen netwerk protocol gebruiken door een rsync daemon te draaien op de file server. Je kunt rsync ook opdragen om ssh te gebruiken, bijvoorbeeld om op een veilige manier veraf gelegen sites te back-uppen. Bekijk de bronnen onderaan voor meer informatie over het opzetten van deze verbindingen.

Alles aan elkaar Scripten

De basisvorm van dit script is afkomstig van de rsync website. In wezen gaat het maar om één commando:

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

De belangrijkste opties zijn:

- *--backup*: maak backups van bestanden vooraleer ze te overschrijven.
- *--backup-dir='date +%Y-%m-%d'*: maak een directory aan voor die bestanden, met een naam die er uitziet zoals: 2004-01-15.
- *-av*: archive mode en verbose mode.

Het volgende script kan elke nacht gedraaid worden via cron (aanwezig in elke distributie). Om het script iedere avond om 23u te laten starten, doe je "crontab -e" en typ je het volgende:

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

Het Script

Hier is mijn shell scriptje dat alles aan elkaar lijmt. Nogmaals, er zijn ook andere manieren om dit aan te pakken, dit is slechts 1 mogelijke implementatie.

```
#!/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=jouw_email_adres_hier

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

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

#####
# 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 for backup stuff
#####
ARCHIVEROOT=directory_om_naar_te_backuppen

#####
# 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_tm
    echo >> /tmp/rsync_script_tmpfile
    echo "#####">>/tmp/rsync_script_tmpfil
    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

```

Bronnen

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

Site onderhouden door het LinuxFocus editors team	Vertaling info:
--	-----------------

© Brian Hone

"some rights reserved" see linuxfocus.org/license/
<http://www.LinuxFocus.org>

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

en --> nl: Floris Lambrechts <floris(at)linuxfocus.org>