

rsync: El mejor sistema de copias de seguridad existente



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



About the author:

Brian Hone es un administrador de sistemas y desarrollador de software de la corporación E Ink. En su tiempo libre disfruta del surfing en agua fría y se desprende por las paredes de las rocas.

Abstract:

La copia de seguridad es una de las partes más duras y más desatendidas de un administrador de sistemas. Es, también una de las más importantes. Es la última defensa contra los fallos de hardware, brechas de seguridad, y la mayor de todas: los usuarios finales. Mientras que hay muchos sistemas de copias de seguridad que cuestan muchos miles de dólares, los cuales se archivan en cintas caras usando software propietario, existe una mejor manera: rsync y un arreglo de discos baratos.

El problema

Podría darte una larga lista de razones de porqué es la pesadilla del administrador de sistemas. Si eres un administrador de sistemas, quizás, no las necesites. Algunas de esas razones son: hardware caro que se estropea más de lo que está operacional, software caro que es una pesadilla para mantener, y largas horas perdidas restaurando múltiples versiones de archivos. Hay situaciones peores, normalmente hay muy poca prioridad en la empresa con respecto a las copias de seguridad, hasta ese día inevitable en que se necesite. Si has realizado copias de seguridad/restauraciones, probablemente hayas tenido esta conversación:

Usuario: "He perdido un archivo. Necesito recuperarlo de cualquier forma."

Administrador: "Bien, ¿cómo se llama?"

Usuario: "No lo sé, creo que tenía una 'e' en el nombre."

Administrador: "Bien y, ¿en qué directorio estaba?"

Usuario: "No lo sé, podía estar en uno de estos tres..."

AdminSis: ¿Sabes cuando usaste el archivo por última vez?"

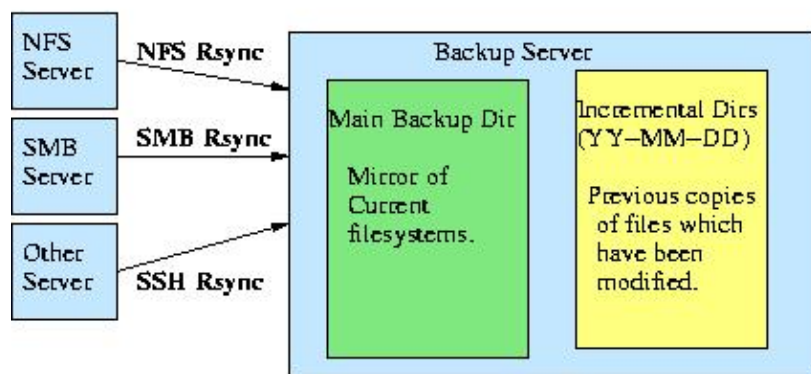
Usuario: "Bueno... creo que fue un martes, entre Febrero o Abril. ¿Cuál es el problema? Creía que teníamos un sistema de copias de seguridad para prevenir este tipo de cosas."

La alternativa, rsync

rsync es una poderosa implementación de un pequeño y maravilloso algoritmo. Su principal poder es la habilidad de replicar eficientemente un sistema de archivos. Usando rsync, es fácil configurar un sistema que mantendrá una copia actualizada de un sistema de archivos usando un arreglo flexible de protocolos de red, tales como NFS, SMB o SSH. La segunda gran característica que consigue este sistema de copias de seguridad es la capacidad de archivar viejas copias de archivos que han sido modificadas o eliminadas. Hay muchas más características de rsync a considerar en este artículo. Recomiendo altamente que lea sobre él en rsync.samba.org.

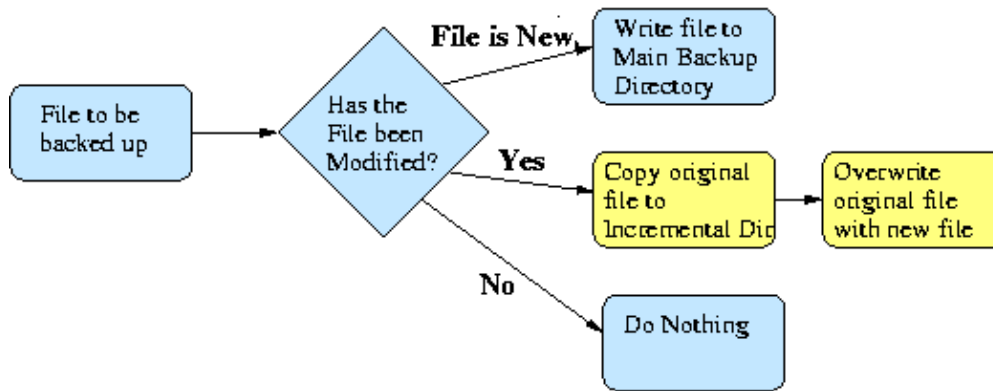
En resumen, este sistema usa una máquina Linux barata con muchos discos baratos y con un pequeño script que llama a rsync. [Fig 1] Cuando se hace una copia de seguridad, le decimos a rsync que cree un directorio llamado 'YY-DD-MM' como lugar para almacenar los cambios incrementales. Seguidamente, rsync examina los servidores de los que hacemos copias de seguridad de los cambios. Si un archivo ha cambiado, se copia la versión vieja al directorio incremental, y luego sobrescribe el archivo en el directorio principal de copias de seguridad. [Fig 2]

Figure 1: Structure of an Rsync Backup Server



En general, los cambios de un día tienden a ser solamente un pequeño porcentaje del total del sistema de archivos. Encuentro que el tamaño medio está entre el 0.5% y 1%. De todas formas, con un conjunto de discos de copias de seguridad que es dos veces el tamaño de nuestros servidores de archivos, puede mantener 50–100 días de copias de seguridad incrementales en el disco duro. Cuando el disco se llene, sólo cambio a un nuevo conjunto de discos, y muevo los viejos. En la práctica, es posible mantener cerca de seis meses de respaldos incrementales en el disco. De hecho, si puede encontrar espacio en cualquier lugar, puede copiar sus incrementales a otro servidor antes de rotar los discos. De este modo, puede mantener un número arbitrariamente grande de incrementales en disco.

Figure 2: Backup of a file using Rsync



Las ventajas: La recuperación de desastres y la restauración de archivos se hace más fácil

Vuelva a la conversación imaginaria de arriba. Ahora, en vez de un incómodo sistema basado en cintas, imagine tener seis meses de copias de seguridad incrementales esperándole en su máquina Linux. Usando su combinación favorita de locate/find/grep, puede encontrar todas la ocurrencias de archivos que posee nuestro usuario imaginario, que contengan una 'e' y que su fecha sea un jueves en Febrero o Abril, y volcarlos dentro de un directorio en el directorio principal del usuario. El problema de averiguar qué versión es la correcta se ha convertido en mi tipo de problema favorito: ¿el de alguien mas?.

Seguidamente, imagine nuestro escenario favorito: fallo completo. Digamos que tiene un gran servidor NSF/SMB que ha perdido. Bueno, si ha hecho copia de seguridad de sus configuraciones, puede levantar su servidor de copia de seguridad como un sustituto solo-lectura en minutos. Intente esto con las cintas.

Porque el uso de rsync para copias de seguridad de disco duro tiene ventaja frente a las cintas

	Cinta de copia de seguridad	rsync
Coste	Muy alto	Bajo
Copia de seguridad completa	Rápido	Rápido
Copia de seguridad incremental	Rápido	Rápido
Restauración completa	Muy lento, probablemente múltiples cintas	Rápido –está todo en disco–
Restauración de archivo	Lento, quizás múltiples cintas, a menudo difícil de encontrar la versión correcta	Muy rápido –está todo en disco y tiene todo el poder de las herramientas de búsqueda de UN*X como find, grep o locate–
Fallo completo		

	La única opción es la restauración completa	Se puede establecer como un servidor de archivos en un momento
--	---	--

Las herramientas

Hay muchas formas para ponerlos en funcionamiento. Todas estas herramientas son de Código Abierto, incluidas en las distribuciones estándar, y muy flexibles. Aquí, describimos una posible configuración, pero no es la única manera.

- **El servidor:** Utilizo Red Hat Linux. Cualquier distribución puede funcionar, al igual que cualquier UNIX. (Incluso lo he configurado con Mac OS X). Una advertencia: mucha RAM ayuda.
- **Disco:** La manera más fácil que hemos encontrado de construir un gran conjunto barato de discos es una tarjeta PCI Firewire conectada a un puñado de discos IDE baratos en cajas Firewire externas. Configurar Linux para usarlos como una gran partición RAID es cosa sencilla.
- **El software:** rsync es una gran herramienta. Es un tipo de herramienta multiuso de replicación de sistemas de archivos. Si no sabe sobre él, compruébelo en rsync.samba.org.
- **Conectando a los servidores de archivos:** rsync es muy flexible. Usamos NFS y smbfs. También puede usar el protocolo de red propio de rsync ejecutando un demonio rsync en el servidor de archivos. También puede decirle a rsync que use SSH para hacer copias de seguridad seguras de sitios remotos. En los recursos de abajo existe información sobre cómo configurar estas conexiones.

Haciendo un script

La forma básica de este script viene del sitio web de rsync. En realidad sólo hay un comando:

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

Los parámetros y opciones son:

- `--backup`: crea copias de seguridad de los archivos antes de sobrescribirlos.
- `--backup-dir=`date +%Y-%m-%d``: crea un directorio de copia de seguridad para esas copias de seguridad, las cuales aparecerán de este modo: 2002-08-15
- `-av`: modo de archivo y modo detallado.

El script siguiente se puede ejecutar cada noche usando la utilidad de Linux cron. Para ejecutar el script cada noche a las 11 pm., utilice el comando "crontab -e", y luego escriba lo siguiente:

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

El script

Aquí está mi script para mantenerlo todo junto. De nuevo, existen muchas formas para hacer esto. Esta es una implementación:

```

#!/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

```

Recursos

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

<p style="text-align: center;"><u>Webpages maintained by the LinuxFocus Editor team</u> © Brian Hone "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org</p>	<p>Translation information: en --> -- : Brian Hone <bhone(at)eink.com> en --> es: Carlos Mayo (homepage)</p>
---	--