

Y es que lo más sencillo es lo mejor, rápido y completo :)

fuelle: <http://danubuntu.wordpress.com/2007/10/27/copia-de-archivos-con-rsync/>

Copia de archivos con rsync

Si quieres tener un respaldo de los archivos del disco duro, *rsync* puede ser muy útil por muchas razones. Está diseñado para disminuir el ancho de banda requerido para la transferencia de archivos, al copiar sólo las diferencias entre los archivos en vez de copiar todo el archivo nuevamente en cada sincronización. Imaginemos copiar un disco duro a diario via LAN. Si lo hiciéramos con *rsync*, tardaría muchísimo menos tiempo. Otra ventaja de *rsync* es que está instalado en prácticamente todos los ordenadores que usan Linux.

- Hay una interfaz para GNOME - *grsync* - que permite usarlo mediante una aplicación gráfica.
- También existe *rdiff-backup* - más funcional que *rsync* para hacer backups (tiene una GUI: *Keep*).

Lo siguiente es básicamente una traducción del [artículo de Joe 'Zonker' Brockmeier](#)

En los dos últimos meses he estado viajando mucho. Durante el mismo período, mi principal computadora de escritorio “sufrió un colapso” y perdí los datos del disco duro. Habría estado en problemas sin *rsync* a mi disposición — pero gracias a mi regular uso de esta utilidad, mis datos (o la mayoría de ellos, más bien) estaban copiados en otro equipo a la espera del momento en que se necesitara. Lleva algún tiempo familiarizarse con *rsync*, pero una lo consigas, podrás manejar la mayor parte de tus necesidades de copia de seguridad con sólo un breve *script*.

¿Qué es lo que hace tan interesante a *rsync*? Para empezar, está diseñado para acelerar las transferencias de ficheros al copiar sólo las diferencias entre ellos en vez de copiar todo cada vez que se haga una copia de seguridad, sincronización, o lo que sea. Por ejemplo, mientras escribo este artículo, puedo hacer una copia via *rsync* ahora y luego otra cuando termine. La segunda (y la tercera, cuarta, etc.) vez que copie el fichero, *rsync* sólo copiará las diferencias. Eso toma mucho menos tiempo, lo que es especialmente importante al hacer copias de grandes estructuras de directorios diariamente. La primera vez puede tardar mucho, pero las siguientes serán mucho más breves.

Haciendo BACKUPS con RSYNC (imprescindible)

Escrito por Dr. Arroyo

Martes, 02 de Septiembre de 2008 05:11 - Actualizado Domingo, 27 de Marzo de 2011 11:03

Otro beneficio es que rsync puede preservar la información de propietarios y permisos, copiar enlaces simbólicos, y que en general, está diseñado para manejar inteligentemente los archivos.

No deberías tener que hacer nada para instalar rsync, pues suele estar disponible en casi cualquier distribución de forma predeterminada. Y si no, lo podrás instalar bajando algún paquete de los repositorios de tu distro. Por supuesto, debes tener rsync en ambas máquinas si vas a copiar datos a un sistema remoto.

Quando lo usas para copiar archivos a otro host, la utilidad rsync normalmente trabaja sobre una shell remota, como *Secure Shell* (SSH) o *Remote Shell* (RSH). Usaré SSH en los ejemplos siguientes, porque RSH no es nada seguro y probablemente no te interese usarlo. También se podría conectar a una máquina remota usando un servidor rsync, pero como SSH es prácticamente omnipresente en estos días, no hay necesidad de molestarse.

Conociendo rsync

La sintaxis básica para rsync es bastante simple — sólo has de ejecutar `rsync [options] origen destino`

para copiar el fichero o ficheros indicados por

origen

argument hasta

destino

.

Así, por ejemplo, si quieres copiar algunos ficheros en tu directorio personal a un dispositivo de almacenamiento USB, podrías usar `rsync -a /home/user/dir/ /media/disk/dir/`. Por cierto, `"/home/user/dir/"` y `"/home/usr/dir"` no son lo mismo para rsync. Sin la barra del final, rsync copia el directorio en su totalidad. Con la barra, copia los contenidos del directorio pero no recrea el directorio. Si estás tratando de reproducir una estructura de directorios con rsync, deberías omitir la barra — por ejemplo, si se quiere replicar `/var/www` en otra máquina o algo por el estilo.

En este ejemplo, he incluido la opción *archive* (-a), la cual combina bastantes opciones de rsync. Combina la copia recursiva con opciones para el manejo de enlaces simbólicos (

Haciendo BACKUPS con RSYNC (imprescindible)

Escrito por Dr. Arroyo

Martes, 02 de Septiembre de 2008 05:11 - Actualizado Domingo, 27 de Marzo de 2011 11:03

symlinks

), preserva el grupo y propietario, y en general, hace de rsync una herramienta muy apropiada para la copia de archivos. Ten en cuenta que no conserva enlaces duros (

hardlinks

); si quieres que haga eso, necesitarás agregar la opción

hardlinks

(-H).

Otra opción que probablemente uses la mayor parte del tiempo es *verbose* (-v), que indica a rsync que muestre información acerca de lo que esté haciendo. Puedes variar la cantidad de información doblando o triplicando la v, de modo que al usar -v obtienes alguna, usando -vv tienes más. y con -vvv rsync te dirá todo lo que pueda.

rsync mueve archivos ocultos (los que comienzan por ".") sin tener que usar ninguna opción especial. Si quieres excluir dichos archivos, puedes usar --exclude=".*/*". Esta opción también te puede ser útil para evitar copiar cosas como los archivos *swap* de Vim (.swp) y copias de seguridad automáticas (como los archivos .bak) creados por algunos programas.

Haciendo copias locales

Supongamos que tienes un disco externo USB o FireWire, y quieres copiar datos de tu directorio *home* al disco externo. Una buena forma de hacer esto sería guardar todos los datos importantes en un solo directorio y luego copiarlo a un directorio de copias de seguridad en la unidad externa usando una orden como:

```
rsync -avh /home/usr/dir/ /media/disk/backup/
```

Si quieres asegurarte de que los archivos que hayas borrado desde la última vez que hiciste esta copia de archivos, **también** se borren del disco externo, puedes añadir la opción --delete, como en este ejemplo:

```
rsync -avh --delete /home/usuario/dir/ /media/disk/backup
```

Haciendo BACKUPS con RSYNC (imprescindible)

Escrito por Dr. Arroyo

Martes, 02 de Septiembre de 2008 05:11 - Actualizado Domingo, 27 de Marzo de 2011 11:03

Ten mucho cuidado con la opción *delete*; con ella, puedes eliminar un montón de ficheros sin querer. De hecho, mientras te acostumbras a usar rsync, probablemente sea una buena idea usar la opción `--dry-run` con las órdenes que vayas a ejecutar, lo que permite hacer un simulacro sin que realmente se copien o sincronicen los ficheros. Si ejecutas una orden y te das cuenta de que esto va a causar alguna forma de daño a los datos, pulsa Ctrl-c inmediatamente para cancelar la transferencia. Puede que hayas perdido algunos ficheros, pero salvarás el resto.

Haciendo copias remotas

¿Y qué pasa si quieres copiar los ficheros a otra máquina de la red? No hay problema — todo lo que necesitas hacer es añadir la información de usuario y host (la máquina remota). Por ejemplo, si quieres copiar el mismo directorio de antes a un equipo remoto, has de hacer esto:

```
rsync -avhe ssh --delete /home/usuario/dir/ usuario@remote.host.com:dir/
```

Si quieres saber lo rápido que va la transferencia, y cuánto queda por copiar, añade la opción `--progress`:

```
rsync --progress -avhe ssh --delete /home/usuario/dir/ usuario@remote.host.com:dir/
```

Si no quieres que te pregunte una contraseña cada vez que de rsync haga una conexión — y no lo querrás — asegúrate de configurar rsync para acceder mediante una clave SSH en lugar de una contraseña. Para ello, se crea una clave SSH en la máquina local usando `ssh-keygen -t dsa`, y pulsa Intro cuando te pregunte por una *passphrase*. Tras crear la clave, usa `ssh-copy-id -i .ssh/id_dsa.pub user@remote.host.com` para copiar la clave pública al equipo remoto.

¿Y qué sucede si necesitas traer de vuelta a algunos de los archivos copiados usando rsync? Usa la siguiente sintaxis:

```
rsync -avze ssh remote.host.com:/home/usuario/dir/ /local/path/
```

Haciendo BACKUPS con RSYNC (imprescindible)

Escrito por Dr. Arroyo

Martes, 02 de Septiembre de 2008 05:11 - Actualizado Domingo, 27 de Marzo de 2011 11:03

La opción z es para comprimir los datos durante su transferencia. Si el fichero que estás copiando existe en el equipo local, rsync lo dejará sin tocar, y lo mismo pasaría si estuvieras moviendo ficheros a una máquina remota.

Concluyendo con el uso de un script

Cuando sepas qué directorio(s) quieres sincronizar, y qué órdenes necesitas para la sincronización, es fácil meter todo en un simple script. Aquí tienes una pequeña muestra:

```
rsync --progress -avze ssh --delete /home/user/bin/ user@remote.host.com:bin/
```

```
rsync --progress -avze ssh --delete /home/user/local/data/ user@remote.host.com:local/data/
```

```
rsync --progress -avze ssh --delete /home/user/.tomboy/ user@remote.host.com:/.tomboy/
```

Usa la opción --progress si vas a ejecutar rsync de forma interactiva. de lo contrario, no se necesita.

Si miras [la página man de rsync](#) , puede parecerte todo muy confuso. Sin embargo, tras un poco de práctica con rsync, encontrarás que no es difícil configurar y realizar tareas con rsync que te ayudarán a prepararte para el día en que tu disco te falle y necesites acceder a esos datos justo en ese momento.