

Sincronización de Ficheros con Rsync

SINCRONÍZATE

Rsync permite sincronizar datos tanto en un ordenador local como remoto. Incluso puede ser usado con SSH para la encriptación de datos. **POR HEIKE JURZIK**

Rsync permite mantener los datos actualizados y ayuda a controlar los confusos cambios que se producen en las nuevas versiones de cada fichero. La herramienta de sincronización Rsync ofrece opciones que ayudan a administrar las propiedades de ficheros, funciona bien con SSH y es perfecto para la transferencia rápida de grandes volúmenes de datos si el ordenador objetivo ya posee una copia de una versión previa. Rsync comprueba las diferencias entre la fuente y las versiones objetivo de los datos y transfiere solamente aquéllos que hayan cambiado.

Sincronicity

Rsync compara los datos en dos máquinas. La sintaxis genérica para esto es *rsync [opciones] fuente objetivo*. La elección de la fuente y el objetivo es crítica. Es preciso decidir cuidadosamente la dirección donde se va a realizar la sincronización para evitar la pérdida de datos. El cuadro "Datos Unison" presenta una alternativa a esta salida de un solo sentido basada en la herramienta *Unison*, la cual permite sincronizar en ambas direcciones.

Rsync ofrece un apoyo importante para la monitorización de la transferencia de datos: si se invoca a Rsync con la etiqueta *-n* el programa presentará una ejecución de prueba:

```
$ rsync -n dir1/* dir2/
skipping directory dir1/dir1
skipping directory dir1/dir2
skipping non-regular file?
"fichero1.tex"
kap01.tex
kap02.tex
kap03.tex
...
```

Tal y como muestra la salida, Rsync transferirá ficheros y directorios pero omitirá los enlaces simbólicos. La herramienta hace referencia a estos últimos como ficheros *no-estándar* (*non-regular*). Para transferir subdirectorios

Datos Unison

Unison [1] es otra importante herramienta de sincronización, que no sólo sincroniza en ambas direcciones, sino que también soporta sistemas operativos no Linux. Este programa se encuentra disponible para Linux, Windows, Mac OS X y Irix, tal y como puede comprobarse si se echa un vistazo al área de descarga de su sitio web.

Unison puede ejecutarse desde la línea de comandos o desde su GUI. Para sincronizar dos directorios en la máquina local se escribe: *unison -ui text carpeta1 carpeta2*, por ejemplo.

Si difiere el contenido de las dos carpetas, Unison dará a conocer la dirección de sincronización y decirnos si el fichero es nuevo o ha sido modificado.

```
carpeta1  carpeta2
new file →   kap01.
tex [f]
← changed   kap02.
tex [f]
```

Para aceptar esta sugerencia se pulsa la tecla [F]. Si se presenta un conflicto que Unison no puede resolver porque, digamos, las dos versiones del fichero han cambiado desde la última vez que se ejecutó el programa, Unison presenta un signo de interrogación *<-?>* invitando a tomar una decisión. Se puede escribir *< ó >* o pulsar [Shift-7] (/) para posponer la decisión hasta otro momento, o pulsar [Shift-l] para posponerla para siempre.

Para ver los comandos disponibles puede escribirse un signo de interrogación (?). Al final, el programa comprueba una vez más si realmente se desean aplicar los cambios: pulsando [Y], se arranca la operación de sincronización.

```
Proceed with?
propagating updates? [ ] y
Propagating updates
UNISON started propagating ?
```

```
changes at 20:20:34 ?
on 28 Feb 2006
[BGN] Copying fichero1.tex
from /home/huhnix/dir1
to /home/huhnix/dir2
...
```

```
Saving synchronizer state
Synchronization complete
Igual de simple es la sincronización con máquinas remotas. Primero hay que asegurarse de que la versión Unison instalada es la misma en ambas máquinas y luego se escribe unison -ui text carpeta1 ssh://usuario@maquinaobjetivo/carpeta.
```

Como puede observarse, la sintaxis es un poco distinta a la de Rsync. En lugar de los dos puntos se tiene una barra. Para usar una ruta absoluta, es preciso añadir otra barra: *unison -ui text carpeta1 ssh://usuario@maquinaobjetivo/carpeta*.

recursivamente hasta el nivel más bajo, habrá que especificar la opción `-r`. Adicionalmente, si se añade la etiqueta `-l` recoge los enlaces simbólicos. Evidentemente es posible una combinación de estas opciones.

```
rsync -rl dir1/* dir2/
building file list ... done
fichero1.tex ->
fichero1_nuevo.tex
kap01.tex
...
```

Rsync dispone de una manera alternativa de manejar enlaces simbólicos: si se sustituye `-l` por `-L`, el programa resolverá el enlace, y los enlaces simbólicos acaban como ficheros normales en el directorio objetivo.

Barra Libre

Al añadir una barra al nombre del directorio se modifica el modo en el que Rsync desarrolla una operación. Por ejemplo, si se da al comando `rsync -a fuente/carpeta objetivo`, Rsync transferirá el directorio *carpeta* y su contenido al directorio objetivo.

Si se añade una barra al directorio objetivo, `rsync` se limitará a transferir el contenido de *carpeta* al objetivo, metiéndolos directamente en el directorio objetivo (Figura 1).

Volver a Empezar

Si se usa Rsync para crear copias de seguridad, tiene sentido mantener los atributos de los ficheros originales. Los atributos incluyen permisos de lectura, escritura y ejecución junto con marcas temporales, es decir, información de la última hora y fecha de acceso (`atime`), del último cambio de estado (`ctime`) y la última modificación (`mtime`).

Además, los administradores pueden aprovechar los parámetros que preservan los datos del propietario y del grupo y soportar ficheros de dispositivo. Para mantener los permisos se especifica la opción `-p`, `-t` se encarga de las marcas temporales y `-g` mantiene intacto el atributo del grupo.

Si bien cualquier usuario normal puede especificar los parámetros mencionados, las opciones `-o` (guarda los datos propietarios) y `-D` (atributos de dispositivo) sólo se encuentran disponibles para el superusuario. La línea de comandos completa con todas estas opciones sería algo como `rsync -rlptgoD home/huhn/*backup/`. Pero existe un atajo: Rsync dispone de una opción especial que combina estos parámetros para este caso. En vez de `-rlptgoD` se escribe `-a`.

Exclusiones

Rsync dispone de una práctica opción que permite excluir determinados ficheros del proceso de sincronización. Para utilizarlo, se especifica la opción `—exclude=` y una pauta de búsqueda y se definen los ficheros a excluir. Esta opción permite el uso de comodines: `rsync -a —exclude=*bak ~/article/*backup/`.

Esto excluye ficheros que acaban en `.bak`. Si se desean excluir ficheros acabados en tilde se añade otro `—exclude:` `rsync -a —exclude=*bak —exclude=*~ fuente objetivo`.

Puede ahorrarse tiempo almacenando las exclusiones en un fichero de texto. Se requiere una línea de separación para cada modelo de búsqueda. Hay que especificar el parámetro `—exclude-from = Fichero de Exclusion` para analizar el fichero.

Cooperación

Rsync incluso puede sincronizar datos a través de una red. Se pasa el parámetro `-e ssh` para encriptar y transferir ficheros usando SSH. Es preciso cambiar la fuente y el objetivo con el nombre o dirección IP de las máquinas local y remota. Si el nombre de la cuenta de la máquina remota es distinta, hay que añadir el nombre de usuario remoto seguido de un signo arroba (`@`), de la siguiente manera: `rsync -e sshb fuente usuario@hostobjetivo:objetivo/`.

Se escribe una ruta relativa al propio directorio de inicio o bien una ruta absoluta después de los dos puntos. Una variable de entorno evita tener que introducir los parámetros para la encriptación SSH cada vez que se ejecuta el comando. Para Bash el comando es: `RSYNC_RSH=ssh`.

Debido a que la variable solamente es válida dentro del contexto de la shell actual hay que añadir la línea de exportar el fichero de configuración de tu Bash, `~/bashrc`, asignar la variable permanentemente y luego ejecutar `source ~/bashrc` para reanalizar el fichero.

Amontonar Datos

Rsync es perfecto para la transferencia de grandes volúmenes de datos. Si se especifica el parámetro `—partial` y se interrumpe la transferencia por alguna razón, puede tomarse dicha transferencia a partir del punto

La barra es importante: ¿Copiará Rsync los contenidos o el fichero en sí?

en la que se dejó. Especificando la opción `—progress` se obtiene un indicador de progreso que permite seguirle la pista de la operación de transferencia.

```
$ rsync --progress --partial
debian.iso transpluto:
/scratch/debian/
Password:
debian.iso
22543992 3% 5.15MB/s
0:02:14
```

Al otro lado de la conexión, el fichero parcial se mantiene oculto en el directorio objetivo. Si se escribe `ls -a`, se puede ver un fichero con un nombre como `.debian.iso.wtKNr7`. El punto al comienzo del nombre del fichero mantiene al fichero oculto y la extensión arbitraria minimiza el peligro de sobrescritura de los ficheros existentes.

Cuando se completa la transferencia, el fichero recobra su nombre original. Si se interrumpe puede ser restablecida especificando nuevamente la opción `—partial`.

La opción `—partial` tiene una desventaja. Si se usa Rsync para actualizar un fichero existente y se interrumpe la transferencia, desaparece el fichero original de la máquina objetivo y es sustituido por el fichero parcial que se ha transferido. Este problema puede solucionarse creando un enlace duro al fichero original primero: `ln debian.iso debian_orig.iso`. Si la transferencia falla, no se perderá la imagen ISO. En vez de ello, aparecerá un nuevo nombre al fichero parcial, pero sin destruir el original.

RECURSOS

[1] Página de inicio de Unison: <http://www.cis.upenn.edu/~bcpierce/unison/>