

Lidiando con grandes ficheros en una red de área local.

Trabajamos con máquinas virtuales con VirtualBox como hipervisor y ya las imágenes ISO de los sistemas operativos son grandes (de 1 a 10 Gigabytes) pero acá hablaremos de máquinas virtuales completas, sus *instantáneas de volúmenes* ("*snapshots*"), vídeos, etc.

Estamos hablando de entre 25 y 50 gigabytes, una cantidad monstruosa si se tiene una red de área local de 100 mbps (*Fast Ethernet*). Incluso teniendo *Gigabit Ethernet* (1000 mbps) nuestros discos duros SATA tampoco tendrán la velocidad suficiente de escritura, por lo que necesitaremos discos de estado sólido *SSD*. Veamos nuestro enfoque.

Esquema de funcionamiento de una Red de Área Local conectada a Internet por medio de un módem y un enrutador inalámbrico.

Compartiendo con SAMBA

Como VirtualBox puede trabajar con diferentes sistemas operativos, muchos de nuestros clientes tienen ambientes mixtos y para compartir archivos entre las diferentes máquinas utilizamos Samba. Hasta acá todo bien, pero el punto práctico es que **una vez hallamos apagado la máquina virtual** podemos "copiar" dicha carpeta al Samba mediante los *enlaces suaves* o *enlaces simbólicos* ("*soft links*" o "*symbolic links*") antes de iniciar el proceso de copia.

Para ello usamos el comando **ln** que crea una especie de redirección cuando cualquier programa lee la carpeta "copiada" (los enlaces duros no permiten trabajar con carpetas), todo esto lo hace de manera transparente el GNU/Linux.

Hay que prestar atención a que si movemos o borramos el directorio fuente, un enlace suave debe ser creado de nuevo, ojo con eso.

Para realizar esto, colocamos este ejemplo y coloreamos para visualizar mejor:

```
ln -s archivo_original.extension mi_enlace_suave_otro_nombre.extension
ln -s /via/arch.txt /otra_via/arch.txt
ln -s /via/carpeta1/ /otravia/carpeta2/
```

Para eliminar un enlace suave simplemente usamos el comando **rm** sobre el enlace suave a borrar. *No importa si el enlace suave se refiere a una carpeta, por lo que al comando **rm** no es necesario el parámetro **-r***, en este caso muy particular.

Software para comprimir

De nuevo, como estamos trabajando en múltiples plataformas, el formato **zip** es el más recomendable para la tarea. Si no lo tenemos instalado pues ejecutamos:

```
sudo apt install zip unzip
```

Comprimiendo una carpeta y su contenido

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>

Suena redundante y lo es: queremos copiar absolutamente todos los archivos y subcarpetas con sus correspondientes ficheros y que se empaquete todo en un solo archivo "comprimido", guardando debidamente las rutas originales.

Ejecutamos:

```
zip -r nombre_archivo.zip carpeta/
```

Incluso podemos comprimir varias carpetas en un mismo archivo "comprimido", ya sea que estén en la carpeta donde estemos trabajando o en otras vías distintas:

```
zip -r nombre_archivo.zip carpeta1/ carpeta2/  
zip -r nombre_archivo.zip /via/carpeta1 /otravia/carpeta2
```

Esto también funciona si es un enlace suave de carpeta:

```
zip -r nombre_archivo.zip carpeta1_suave carpeta2  
zip -r nombre_archivo.zip /via/carpeta1_suave /otravia/carpeta2
```

Y, por supuesto, el archivo comprimido en sí mismo lo podemos crear en cualquier otro sitio que necesitemos:

```
zip -r nombre_archivo.zip carpeta1_suave/ carpeta2/  
zip -r nombre_archivo.zip /via/carpeta1_suave/ /otravia/carpeta2/
```

Comprobando un archivo comprimido

Es muy importante comprobar **antes de copiar fuera del equipo origen** si el archivo comprimido ha sido creado correctamente. Para ello usaremos el comando **-T** y lo acompañamos del

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.
<https://www.ks7000.net.ve>

parámetro **-v** para ver en detalle el contenido:

```
zip -T -v nombre_archivo.zip
```

¿Vale la pena comprimir?

No, no vale la pena. Son tal cantidad de datos que tardaría mucho tiempo y esfuerzo el comprimir las carpetas y sus archivos, para evitar esto usamos el parámetro **-0** (un guion y un cero):

```
zip -r -0 nombre_archivo.zip carpeta/
```

Comentarios

Pues eso, introducir comentarios en el fichero comprimido, pues colocamos el parámetro **-z**:

```
zip -r -0 -z nombre_archivo.zip carpeta/
```

Al presionar la tecla Intro y cuando termine de "comprimir" pedirá el mensaje que puede ser de varias líneas, pero la última línea solo debe contener un punto y presionamos Intro para así terminar de guardar nuestra obra. El mensaje se visualizará cuando verifiquemos el comprimido, según mostramos en una sección anterior.

Identificador de archivo según *md5sum*

La idea es obtener un *hash* o identificador digital que garantice que el fichero que copiaremos es idéntico al que creamos de manera "comprimida". Aunque el programa **md5sum** está instalado por defecto en la mayoría de las distribuciones GNU/Linux, podemos comprobar si tenemos completamente instalado **coreutils**:

```
sudo apt install coreutils
```

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>

```
sudo apt show coreutils
```

Luego invocamos sobre el archivo que vamos a copiar:

```
md5sum fichero_comprimido.zip > md5_fichero_comprimido.txt
```

Y cuando hayamos terminado de copiar y rearmar en la otra máquina le calcularemos allá su *hash* y deben coincidir. Esto lo haremos porque, como dijimos, vamos a trocear el gigantesco archivo en partes manejables.

Dividiendo archivos, comando *split*

El comando **split** también forma parte del super paquete **coreutils** por lo que pasamos a usarlo de una buena vez:

```
split nombre_archivo.zip -b 1000m -d "comprimido.z"
```

- **split** : el comando que divide.
- **nombre_archivo.zip** : el fichero "comprimido" que queremos pasar.
- **-b 1000m** : con este parámetro indicamos que queremos trozos de mil megabytes cada uno.
- **-d** : indicamos que queremos sufijos numéricos a fin de que puedan ser rearmados en el mismo orden.
- **"comprimido.z"** : por último pasamos el prefijo deseado, el cual usaremos como guía para rearmar (o copiar, veamos la próxima sección).

Copiando los trozos

Lo más probable es que usemos algún entorno gráfico (explorador de archivos) para copiar los trozos, pero si usamos la línea de comando estas recomendaciones pueden ser útiles:

- Si utilizamos **rsync**, sin configuración especial alguna, recordemos que copiará todos los archivos, los trozos y el gigantesco "comprimido". Lo más "rápido y sucio" es borrar el gigantesco y el enlace suave, solo dejaremos los trozos y el fichero de texto con el *hash* md5. **rsyn** tiene una opción para controlar la velocidad de copia, eso es bueno para no colapsar la red local; a la fecha no hemos hallado la manera que "reanude" una descarga, pero como los trozos son manejables, poco será el tiempo que perdamos si tenemos que apagar los equipos, o falla la electricidad, etc. **Es muy importante que tengamos presente que rsyn tiene un origen y un destino, recuerden que estamos copiando, no sincronizando.**
- También pudiéramos usar **sftp**, por ahora no tenemos consejo especial acá.
- Aunque ya estamos usando samba, tampoco podemos descartar usar un servidor FTP, además que esta opción es atractiva porque la opción de reanudar "subidas" y "bajadas" es la norma y no la excepción, como nos ocurrió en los años 1990.

Rearmando el "comprimido"

Habiendo conectado por **ssh** usaremos el comando **cat**, otro de **coreutils**. Aunque generalmente usamos **cat** para mostrar el contenido de un fichero, generalmente de texto, su uso principal es para concatenar archivos ("**concatenate**")

```
cat comprimido.z* > comprimido.zip
```

Esto es posible porque están numerados como **comprimido.z01**, **comprimido.z02**, **comprimido.z03**, etc. Una vez esté unida, pues ya sabemos como comprobar con **zip** y obtener su *hash* con **md5sum** y comparar con lo almacenado en el fichero de texto (que veremos con **cat**).

"Descomprimiendo" ficheros

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>

Usaremos **unzip**, considerando que tenemos espacio en la máquina destino, y que lo queremos extraer en la misma carpeta donde rearmamos el "comprimido":

```
unzip comprimido.zip
```

De esta manera se recrearán todas las carpetas, subcarpetas y ficheros necesarios para la máquina virtual que queremos copiar. si necesitamos extraerlos en una carpeta específica:

```
unzip comprimido.zip -d /via_nueva/
```

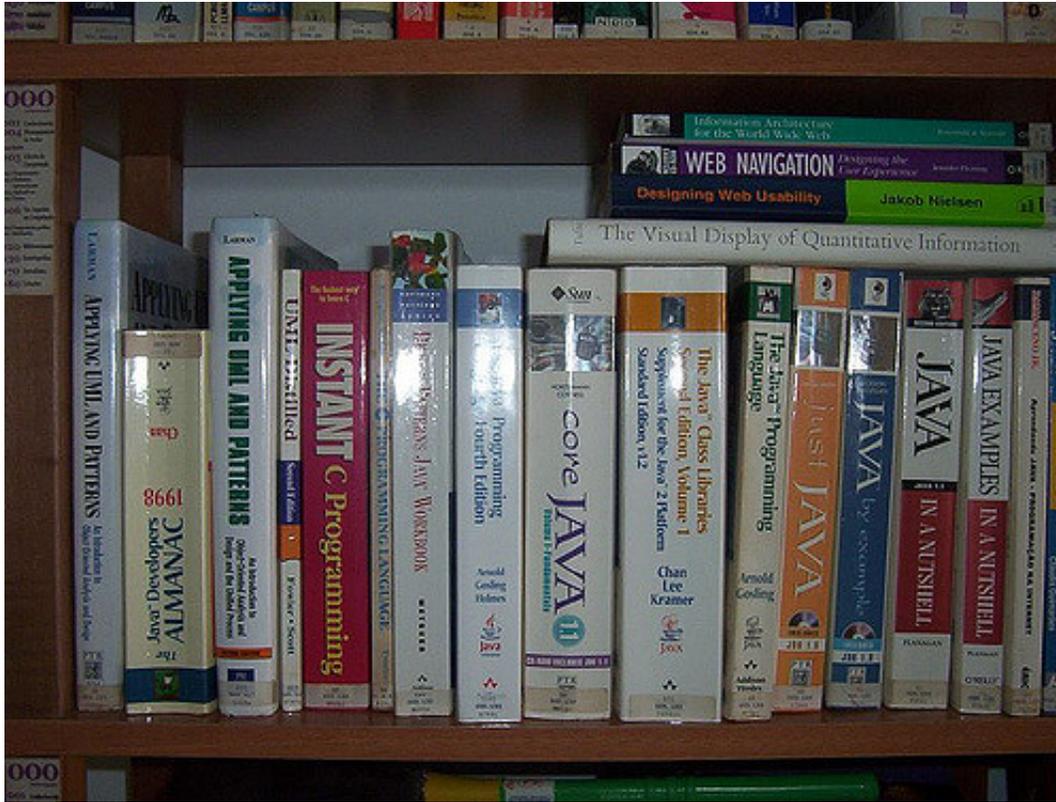
Automatización del proceso

Bajo ciertas circunstancias podremos automatizar el proceso con un guion de procesos pro lotes **pero debemos tener en cuenta el siguiente ambiente, ya que no estamos creado un proceso que examine y tenga en cuenta diferentes factores**. Debe ser algo sencillo con el que le pasemos el nombre de la carpeta local, el nombre de la carpeta remote y *las credenciales del usuario con el cual conectaremos en la otra máquina*, veamos.

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>



Fuentes consultadas

En idioma castellano

- « ».
- « ».
- « ».

En idioma francés

- « ».
- « ».

- «».

En idioma inglés



Beethoven era un buen compositor porque utilizaba ideas nuevas en combinación con ideas antiguas. Nadie, ni siquiera Beethoven podría inventar la música desde cero. Es igual con la informática

(Richard Stallman)

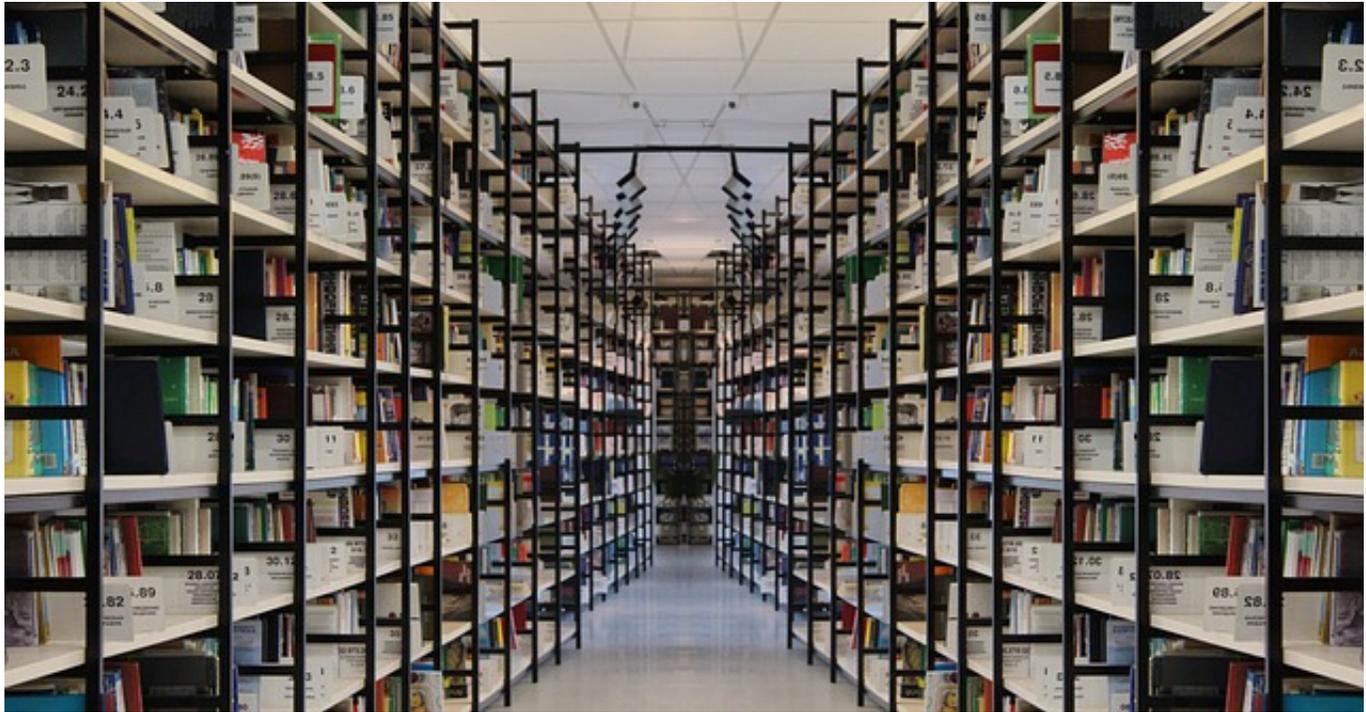
akifrases.com

- [«What is a hard and symbolic \(soft\) link in Linux or Unix?»](#).
- [«How to zip a folder in Ubuntu Linux / Debian Linux»](#).
- [«How to Compress and Split Files in Ubuntu»](#).
- [«HowToMD5SUM»](#).

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>



Crédito de la imagen [Gerd Altmann](#), [trabajo](#), licencia de uso: [Pixabay](#)
