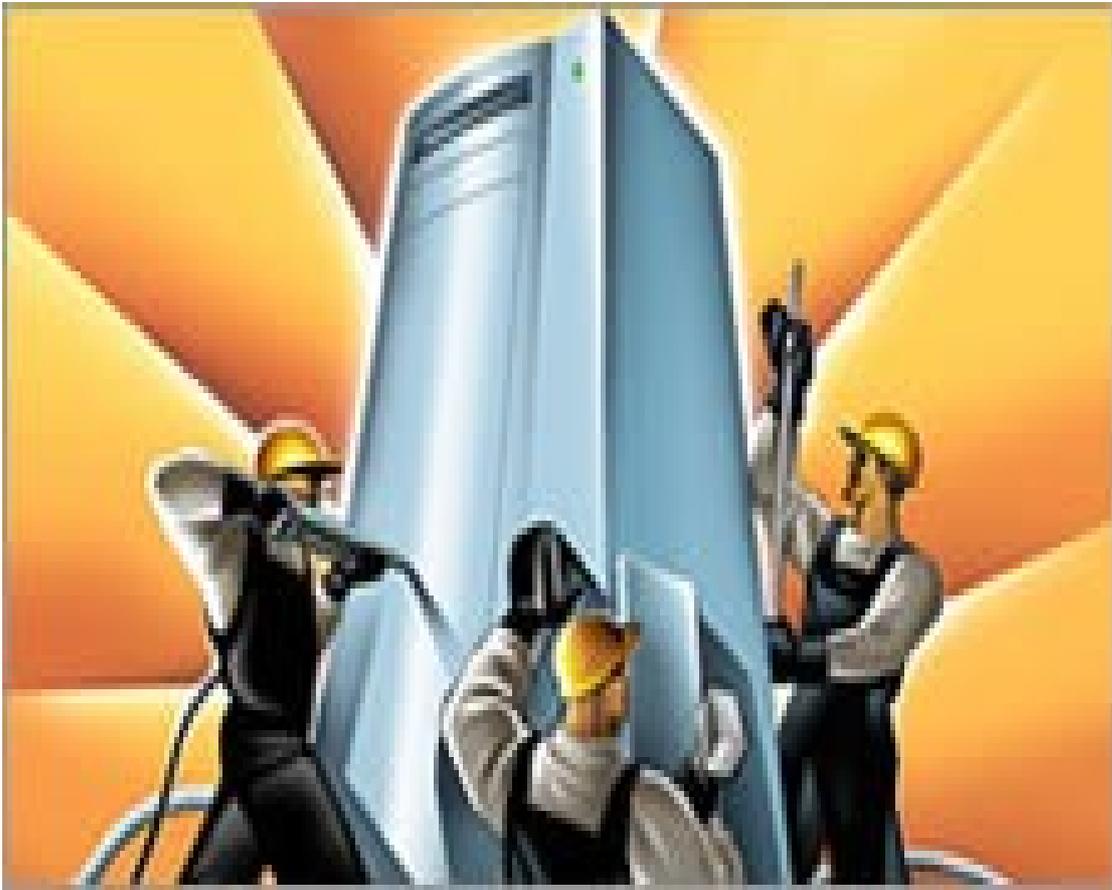


Sistemas de Backups Libres



Diciembre 2008
Curso Linux Nivel I
UADER – Gugler

Gustavo Tomasini

gustavotomasini@yahoo.com.ar

basado en un trabajo de:
AP Christian Saluzzo
ASS Marclo Kogan

Índice de contenido

Índice de contenido

Copyright (c) 2008.....	3
Herramientas libres para realizar Backup.....	3
Bacula	4
Elementos.....	4
Instalación.....	7
Amanda.....	19
Funcionamiento	19
Para qué sirve Amanda	20
Si tengo problemas ¿A quién acudo?.....	20
Un ejemplo práctico.....	20
Rsync.....	40
Algoritmo.....	40
Características y aplicaciones	40
Uso de la aplicación	40
Variantes.....	41
FlyBack 0.4.0	42
Copias de seguridad al estilo TimeMachine de Apple	42
Para utilizar FlyBack necesitas:.....	42
Requisitos mínimos:.....	42
FlyBack utilidad de backup para Linux	42
Backupper 0.24	44
Realiza copias de seguridad periódicas de tus carpetas importantes	44

Copyright (c) 2008 Gustavo Tomasini Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Herramientas libres para realizar Backup

Hemos observado a través de años de trabajo en distintas empresas, que se le da muy poca importancia a algo fundamental para su normal funcionamiento. El Backup.

En reglas generales y en todos los ámbitos, ya sean profesionales o familiares, la copia de seguridad es valorada en un solo momento: El momento en que se pierde toda la información y no está hecha, o está hecha deficientemente.

Hemos investigado sobre la existencia de software que satisfaga las necesidades de los usuarios de esta región:

El software de backup, según hemos visto, debe cumplir con las siguientes características para satisfacer al usuario de nivel medio, usuario familiar, usuario poco experimentado o usuario profesional:

- Debe ser económicamente **muy viable**; ya que cuando hablamos de invertir dinero en un software de backup, la primera respuesta que recibimos es NO.
- No debe requerir mucho hardware, preferentemente que funcione sin tener que invertir en nada de Hardware.
- Debe trabajar en forma **desatendida**. El usuario está demasiado ocupado en infinidad de tareas, y es lógico que así sea, así que no recordará, ni se ocupará **nunca** de ejecutar o controlar los backups, por lo tanto éste se debe realizar sin intervención del usuario.
- Debe ser **estable** y **seguro**, en este punto el software libre, y especialmente el que está basado en tecnologías Unix, se destaca notablemente, y aquí lo que estamos buscando fundamentalmente es **seguridad**.
- Debe ser **fácil de administrar**. Con pocos pasos debe ser fácil controlar para comprobar que se estén realizando las copias eficientemente.
- Debe realizar copias **rápidas**, **sin consumir** muchos recursos y **sin intervenir** en las tareas habituales del usuario, en caso de molestarlo, éste no dudará en desinstalarlo o cerrarlo para seguir con sus tareas habituales.

Ante estos requerimientos, a continuación pasamos a estudiar parte del software libre existente en el mercado para realizar backups.

- Bacula
- Amanda
- Rsync
- FlyBack 0.4.0

– Backerupper 0.24

Bacula

Bacula¹ es una colección de herramientas de respaldo muy amplia, capaces de cubrir eficientemente las necesidades de respaldo de equipos bajo redes IP. Se basa en una arquitectura cliente/servidor que resulta muy eficaz y fácil de manejar, dada la amplia gama de funciones y características que brinda; copiar y restaurar ficheros dañados o perdidos. Además, debido a su desarrollo y estructura modular, Bacula se adapta tanto al uso personal como profesional, para parques de ordenadores muy grandes.

Elementos

Los elementos necesarios para que Bacula funcione son:

- **Bacula-director**

Es el demonio que gestiona la lógica de los procesos de backup y los demás servicios. El servidor de la base de datos debe estar accesible desde la máquina que ejecuta este demonio (o también puede estar en la misma máquina y escuchar en localhost).

En el archivo de configuración de este demonio se especifica dónde y cómo acceder al resto de demonios y recursos, la contraseña para el acceso mediante bacula-console y los trabajos o jobs.

- **Bacula-storage daemon**

Este demonio es el encargado de manejar los dispositivos de almacenamiento; esto exige que este demonio esté instalado en la máquina que posea la conexión física a los dispositivos de almacenamiento, tales como: discos locales, grabadoras de CD o DVD, unidades de cinta, volúmenes NAS o SAN, autocargadores o librerías de cinta.

El fichero de configuración de este servicio define tanto los dispositivos de almacenamiento que maneja, como que directores pueden utilizarlo.

- **Bacula-file daemon**

Mediante este demonio Bacula obtiene los ficheros que necesita respaldar, así pues éste es el componente que hay que instalar en las máquinas que necesiten respaldo. Realiza la misma función que los "agentes" en otros sistemas de backup.

Este archivo de configuración es el más simple de todos, simplemente especifica qué directores pueden realizarle peticiones.

Para poder interactuar con el servicio de backup, necesitaremos un cliente:

- **Bacula-console**

1 <http://es.wikipedia.org/wiki/Bacula>

Todo el conjunto de elementos que forman Bacula trabaja en sincronía y es totalmente compatible con bases de datos como [MySQL](#), [SQLite](#) y [PostgreSQL](#).

Bacula² gestiona la creación de copias de seguridad, la restauración de sistemas y la verificación de datos a través de red.

Se ha diseñado pensando en grandes redes locales con decenas de ordenadores. Su arquitectura cliente - servidor facilita enormemente la labor de programar las copias. Simplemente hay que instalar el servicio en cada ordenador. Después podremos controlar todas las copias desde un solo PC.

Las posibilidades de Bacula son muchas, pero para sacar el máximo rendimiento habría que disponer de un servidor dedicado que se encargue de la tarea de copiar automáticamente los datos que hayamos elegido.

El programa está pensado para manejarse en consola de comandos. No es que esté destinado a profesionales, pero sí a usuarios expertos.

Comprendiendo a bacula ³

Cuesta cinco minutos comprender el funcionamiento del sistema, gracias a la documentación y a la separación lógica de las tareas a realizar. La configuración es algo más compleja.

Queremos hacer una aclaración, aunque hablamos de un sistema de backup en cinta, no quiere decir que unicamente se pueda volcar en ese tipo de dispositivos, se puede utilizar perfectamente en una red doméstica, o en una sola máquina y grabar los respaldos en un CD, pero obviamente bacula está orientado a una red más grande y con uno o varios dispositivos de cinta.

Bacula guarda todas sus operaciones, trabajos, listas de volúmenes, etc en una base de datos estilo SQL. Toda la instalación se ha realizado en sistemas debian, salvo algunos clientes, que utilizan otra serie de distribuciones (Red Hat y Mandrake), así que la instalación de bacula se redujo a un apt-get y el solito creo las bases de datos necesarias.

El sistema se divide en:

- Bacula director El demonio encargado de gestionar todas las operaciones de backup. El director sabe los trabajos que se van a realizar, cuando , donde y como. Y además se encarga de restaurar los ficheros que le pidamos y su verificación (una especie de suma de comprobación de integridad). Se puede instalar en cualquier máquina de la red.
- Bacula File El cliente. Es necesario instalarlo en todas las máquinas de las que queramos hacer respaldo. Su función es leer y transmitir los ficheros que el director le pida, o restaurarlos.
- Bacula Storage Este demonio se encarga de la lectura/escritura física en los volúmenes que estén definidos (cintas, ficheros)

² <http://bacula.softonic.com/linux>

³ <http://libertonia.escomposlinux.org/story/2004/5/5/93545/27085>

Además tenemos la consola, con la cual nos conectaremos al director, y desde donde podremos dar ordenes, hacer consultas, etc. Y el catálogo, donde el director guarda y registra todas sus operaciones.

Aclarado esto, conviene también explicar que son los volúmenes y los "pools":

Bacula se refiere a los volúmenes como los dispositivos físicos donde se guardan los volcados (ficheros o cintas) y a los "pools" como el conjunto de uno o varios volúmenes. Para que quede más claro, nosotros definimos por ejemplo un "pool" que va a servir para hacer copias diarias de diferentes máquinas, y en ese "pool" añadimos diferentes volúmenes y bacula sabe cuales volúmenes pertenecen a cada "pool". De esta manera, bacula conoce donde está cada cosa, y de donde sacarla, o bien donde será necesario escribir cuando el volumen actual esté lleno.

El funcionamiento viene a ser algo así:

El director, que tiene definidos una serie de trabajos y a una determinada fecha y hora se pone en marcha, contacta con el primer cliente que tiene definido, el cliente de esa máquina comprueba los ficheros o directorios que le pide el director y le devuelve los que hayan cambiado (si es una copia diferencial o incremental) al director, este se pone en contacto con el demonio "storage" que almacena en el volumen los ficheros que le mandan, y vuelta a empezar por cada uno de los clientes. Para un trabajo de verificación viene a ser lo mismo, solo que no se guardan los ficheros, sino solo las sumas de comprobación en la base de datos SQL que usemos (a día de hoy SQLite, MySQL o PostgreSQL).

Instalación

Instalación a partir de binarios:

- Si el kernel no viene configurado por defecto es necesario recompilar kernel de la máquina que tenga conectado el dispositivo de almacenamiento(cintas, CD, DVD, Discos Duros, etc.:
 - <*> SCSI support
 - SCSI support type (disk, tape, CD-ROM)
 - <*> SCSI disk support
 - (40) Maximum number of SCSI disks that can be loaded as modules
 - <M> SCSI tape support
- Los daemons: Al trabajar en red, la base datos puede estar en cualquier parte, el director también, los clientes cada uno en su máquina y el daemon que trata con las cintas o discos en la máquina que tiene las cintas o discos. Lo ideal es no complicarse la vida e instalar el director, la base de datos y el daemon encargado de grabar datos en la máquina con el scsi de cintas o discos. Y la consola en el ordenador normal de trabajo (aunque al instalar el cliente en debian, se te instala también la consola)

Entonces tenemos lo siguiente:

```
maquina@directora:~#apt-get install bacula-director-mysql bacula-sd
maquina@cliente:~#apt-get install bacula-fd
```

Una vez que está instalado y configurado, se debe retocar los archivos de configuración. Todos los archivos de configuración son bastante simples y apenas requieren alguna modificación, salvo el del daemon director que es bastante extenso. En realidad, en las últimas versiones de bacula se ha simplificado bastante este archivo ya que se ha implementado un recurso que permite tener definidos una serie de trabajos modelo, que despues se pueden modificar puntualmente para cada cliente/trabajo. Viene a ser algo parecido a las clases de la programación orientada a objetos. Quedando un archivo así:

```
maquina@directora:~# wc -l /etc/bacula/bacula-dir.conf
1099 /etc/bacula/bacula-dir.conf
```

Los archivos de configuración de bacula

Ahora hablaremos de los diferentes archivos de configuración de Báculo.

Describiremos desde los más fáciles a los más difíciles:

- **/etc/console.conf**

La consola de bacula es el medio por el cual nos vamos a comunicar con el sistema director. Su archivo de configuración es este:

```
gandalf:/etc/bacula# cat console.conf
#
# Bacula User Agent (or Console) Configuration File
#

Director {
  Name = maquinadirectora-dir
  DIRport = 9101
  address = maquinadirectora.x.com
  Password = "clave"
}
```

El nombre ha de ser el mismo que más tarde definiremos en el archivo de configuración del

director. Lo siguiente es el puerto donde escucha el director. El FQDN o bien su dirección IP. Y finalmente el password que se necesita para conectar al director.

- **/etc/bacula-fd.conf**

El archivo de configuración de los clientes

```
gandalf:/etc/bacula# cat bacula-fd.conf
#
# List Directors who are permitted to contact this File daemon
#
Director {
  Name = maquinadirectora-dir
  Password = "clave"
}

#
# "Global" File daemon configuration specifications
#
FileDaemon {
  Name = gandalf-fd          # this is me
  FDport = 9102             # where we listen for the director
  WorkingDirectory = /var/lib/bacula
  Pid Directory = /var/run/bacula
}

# Send all messages except skipped files back to Director
Messages {
  Name = Standard
  director = maquinadirectora-dir = all, !skipped
}
```

Basicamente, se autoriza al director a conectar con nosotros por el puerto 9102. Definimos el nombre de nuestro cliente, y se definen el tipo de mensajes que queremos hacer llegar al director.

- **/etc/bacula-sd.conf**

El daemon encargado de escribir en los dispositivos de cinta tiene su propio archivo:

```
maquinadirectora:/etc/bacula# cat bacula-sd.conf
# You may need to change the name of your tape drive
# on the "Archive Device" directive in the Device
# resource. If you change the Name and/or the
# "Media Type" in the Device resource, please ensure
# that dird.conf has corresponding changes.
#

Storage {
  Name = maquinadirectora-sd          # definition of myself
  SDPort = 9103                      # Director's port
  WorkingDirectory = "/var/lib/bacula"
  Pid Directory = "/var/run/bacula"
}

#
# List Directors who are permitted to contact Storage daemon
#
Director {
  Name = maquinadirectora-dir
  Password = "clave"
}

#
```

```
# Devices supported by this Storage daemon
# To connect, the Director's bacula-dir.conf must have the
# same Name and MediaType.
#

Device {
  Name = DLT4-0
  Media Type = DLT4
  Archive Device = /dev/nst0
  LabelMedia = Yes;          # lets Bacula label unlabeled media
  RandomAccess = yes;
  AutomaticMount = yes;     # when device opened, read it
  RemovableMedia = yes;
  AlwaysOpen = no;
}

Device {
  Name = DLT4-1              #
  Media Type = DLT4
  Archive Device = /dev/nst1
  AutomaticMount = yes;     # when device opened, read it
  LabelMedia = Yes;
  AlwaysOpen = no;
  RemovableMedia = yes;
  RandomAccess = yes;
}

#
# Send all messages to the Director,
# mount messages also are sent to the email address
#
Messages {
  Name = Standard
  director = maquinadirectora-dir = all
}
```

-P. En este archivo se definen varios recursos:

1. Storage

Definimos el nombre del daemon y el puerto de escucha

2. Director

Al igual que en los otros, el nombre del director que se nos va a conectar, con su password.

1. Devices

Definimos los dispositivos físicos de escritura. Cintas , CD, DVD, Discos Duros ect. En este recurso vemos que le damos un nombre al dispositivo, indicamos el tipo de medio, y el nombre físico tal y como se ve desde el sistema de archivos. Lo demás son diferentes opciones, como que se monten automáticamente los dispositivos o que se permitan etiquetar.

3. Messages

El tipo de mensajes que queramos recibir.

- **/etc/bacula-dir**

Este archivo es excesivamente largo, comentaremos sus secciones por orden.

#Definición del director

```
Director {
    # define myself
    Name = maquinadirectora-dir
    DIRport = 9101 # where we listen for UA connections
    QueryFile = "/etc/bacula/scripts/query.sql"
    WorkingDirectory = "/var/lib/bacula"
    PidDirectory = "/var/run/bacula"
    Maximum Concurrent Jobs = 1
    Password = "clave" # Console p$ Messages = Standard
}
```

Definimos el nombre del director, su puerto de escucha y la clave. Además vemos una opción que permite fijar cuantos trabajos se realizarán simultáneamente. En la documentación se recomienda poner a 1.

```
Job {
    Name = cliente1-diaria
    Client = cliente1-fd
    Type = backup
    Level = Incremental
    FileSet = Diaria
    Schedule = Diaria
    Messages = Standard
    Pool = Diaria

    Storage = tape1
    # JobDefs = "Diaria"
}
```

Lo descripto arriba es una definición completa de un trabajo. En donde definimos su nombre, el nombre del cliente, el tipo de trabajo, su nivel, los archivos a los que se accede, el calendario, los mensajes que manda, el pool a utilizar y el dispositivo de escritura. La mayoría de estos campos los definiremos en este mismo archivo más adelante. Se debe escribir esta definición por cada tipo de trabajo y cliente. En versiones recientes se ha simplificado mucho este problema al permitir definir un modelo de trabajo que luego se utilizaba en todas las definiciones.

```
#Backup diario
JobDefs {
    Name = Diaria
    Type = Backup
    Level = Incremental
    FileSet = Diaria
    Schedule = Diaria
    Messages = Standard
    Pool = Diaria
    Storage = tape0
}
```

Hacemos una sola definición por cada tipo de trabajo y luego:

```
Job {
    Client = dell1-fd
    JobDefs = "Diaria"
}
```

Pasamos al siguiente recurso

```
FileSet {
    Name = "Diaria"
    Include = signature=MD5 {
        /var
        /etc
        /root
        /home
    }
}
```

```

    Exclude = { /proc /tmp /.journal /.fsck /var/run }
}

```

Le decimos que directorios o archivos queremos que nos respalde, incluyendo una suma md5 y cuales queremos excluir. Y por supuesto le damos un nombre que lo identifique en los trabajos

```

Schedule {
    Name = "Diaria"
    Run = Incremental mon-sat at 20:00
}

```

Los calendarios de los trabajos y su tipo, aunque no es necesario indicarlo porque ya lo tengo definido en los trabajos. Se pueden hacer muchas combinaciones con las fechas y las horas, que ejecute un trabajo el tercer jueves de cada mes, el último domingo de Mayo, etc. En el ejemplo realiza un trabajo de lunes a sábado a las 20:00 horas

```

Client {
    name = cliente1-fd
    Address = cliente1.x.com
    FDPort = 9102
    Catalog = MyCatalog
    Password = "clave"
    File Retention = 30 days      # 30 days
    Job Retention = 6 months     # six months
    AutoPrune = yes             # Prune expired Jobs/Files
}

```

Pasamos a la definición de los clientes, se necesita una de estas por cada uno de ellos. Le ponemos un nombre, que será obligatoriamente el que ya pusimos en el daemon que se ejecuta en la máquina a respaldar, su dirección en la red, el puerto donde escucha, el catálogo (la base de datos), su clave, el tiempo que retendrá los archivos guardados, el tiempo que quedará registrado el trabajo en la base de datos, y si queremos que se borren automáticamente los registros de la base de datos pasado el tiempo definido.

```

Storage {
    Name = tape0
    Address = maquinadirectora    # N.B. Use a fully qualified name here
    SDPort = 9103
    Password = "clave"
    Device = DLT4-0
    Media Type = DLT4
}

```

En esta parte nombramos los recursos para conectar con el daemon de almacenamiento. Le damos un nombre, su dirección en la red, el puerto de escucha y su clave de acceso. Además le indicamos el dispositivo y tipo de medio.

```

Catalog {
    Name = MyCatalog
    User = bacula
    dbname = bacula
    password = "clave"
}

```

El recurso relacionado con la base de datos que creamos previamente a la instalación de bacula. El nombre, el usuario permitido en la base, la base misma y el password.

```

Messages {
    Name = Standard
    mailcommand = "/usr/lib/bacula/smtp -h localhost -f \"\\(Bacula) %r\" -s
    \"Ba$ operatorcommand = "/usr/lib/bacula/smtp -h localhost -f \"\\(Bacula)
    %r\" -s $ mail = root@localhost = all, !skipped
    operator = root@localhost = mount
    console = all, !skipped, !saved
}

```

```
append = "/var/lib/bacula/log" = all, !skipped  
}
```

Los mensajes que genera el director y a donde mandarlos, se configuran aquí. Tal y como está puesto, genera un mensaje por trabajo (bastante molesto), pero es bueno porque uno se entera de todo lo que sucede.

```
Pool {  
  Name = Diaria  
  Pool Type = Backup  
  Recycle = yes           # Bacula can automatically recycle Volumes  
  AutoPrune = yes        # Prune expired volumes  
  Volume Retention = 100 days  
  Accept Any Volume = yes # write on any volume in the pool  
}
```

Bácula en ejecución

Entramos de lleno en el funcionamiento de Bácula, y los errores más frecuentes que podemos encontrar

Una vez que hemos configurado bacula, podemos comprobar que los archivos estén correctamente con la opción `-t` de los tres daemons:

```
maquinadirectora:~#bacula-dir -t /etc/bacula-dir.conf
```

Si tenemos un error en la configuración nos lo dirá. Debemos tener en cuenta que solo nos dirá si el error es de sintaxis, problemas de conexión a causa de la red, o errores en los passwords/nombres tendremos que averiguar la causa por nuestra cuenta.

Para arrancar el programa nos vamos a la maquinadirectora y ejecutamos si no estaban ya corriendo:

```
maquinadirectora:~# /etc/init.d/mysql start
maquinadirectora:~# /etc/init.d/bacula-sd start
maquinadirectora:~# /etc/init.d/bacula-fd start
maquinadirectora:~# /etc/init.d/bacula-dir start
```

Arrancamos también cada uno de los clientes que vayamos a utilizar, y si todo ha ido bien, en ninguna de las máquinas habremos obtenido un error. En este momento lo único que puede haber sucedido es que tengamos algún error sintáctico en alguno de los archivos, y el daemon se niegue a arrancar. Lo solucionamos y pasamos al siguiente paso, que es etiquetar los dispositivos que vayamos a usar y añadirlas a los pools que tengamos definidos. Aquí ya entraría en juego la estrategia de cada administrador a la hora de establecer una política de backups. Hay que tener en cuenta la importancia de los datos, la cantidad de cintas o discos que tenemos etc, para organizarlo todo. En la documentación de Bácula hay una sección referente a ello que ayuda a establecer esa política. Por ejemplo si se definen tres pools: Diaria con copias incrementales, Semanal con copias diferenciales, Mensual con copias completas, y un trabajo diario de verificación de sumas de los archivos. Una vez tengamos esto claro y distribuidas las cintas o discos por los diferentes pools hay que etiquetarlos físicamente (para que pueda distinguirlas el usuario) y por software (para que Bácula las reconozca). Ejecutamos la consola de bacula:

```
gandalf:~# bconsole
Y obtenemos el prompt del director:
Connecting to Director maquinadirectora.x.com:9101
1000 OK: directora-dir Version: 1.32f-5 (09 Mar 2004)
Enter a period to cancel a command.
*
```

Desde aquí tenemos pleno control del director y ahora es cuando podemos etiquetar ejecutando el comando **label** Que nos preguntará por los diferentes dispositivos que tengamos configurados. Para el ejemplo, las dos bocas de la unidad de cintas

```
The defined Storage resources are:
 1: Tape0
 2: Tape1
Select Storage resource (1-2):
Enter new Volume name:
```

Le ponemos un nombre y si no ha sido ya definido previamente, nos preguntará por el nombre del pool al que queremos añadir la cinta. Una vez se lo facilitemos, bacula lo tendrá marcado para solicitarnoslo cada vez que lo necesite. Repetimos la operación por cada uno de los volúmenes a

utilizar. Una vez terminado ejecutamos *list media* y nos deberían salir todos los pools definidos con sus respectivos volúmenes, bytes escritos, última escritura y un sinfín de datos más.

Ya que tenemos esto definido, podemos pasar ahora a comprobar el estado de los diferentes daemons. Si nos hemos conectado al director es evidente que está en marcha, pero no sabemos mucho de los clientes ni del daemon de almacenamiento. Para esto utilizamos las ordenes *status client* y *status storage*

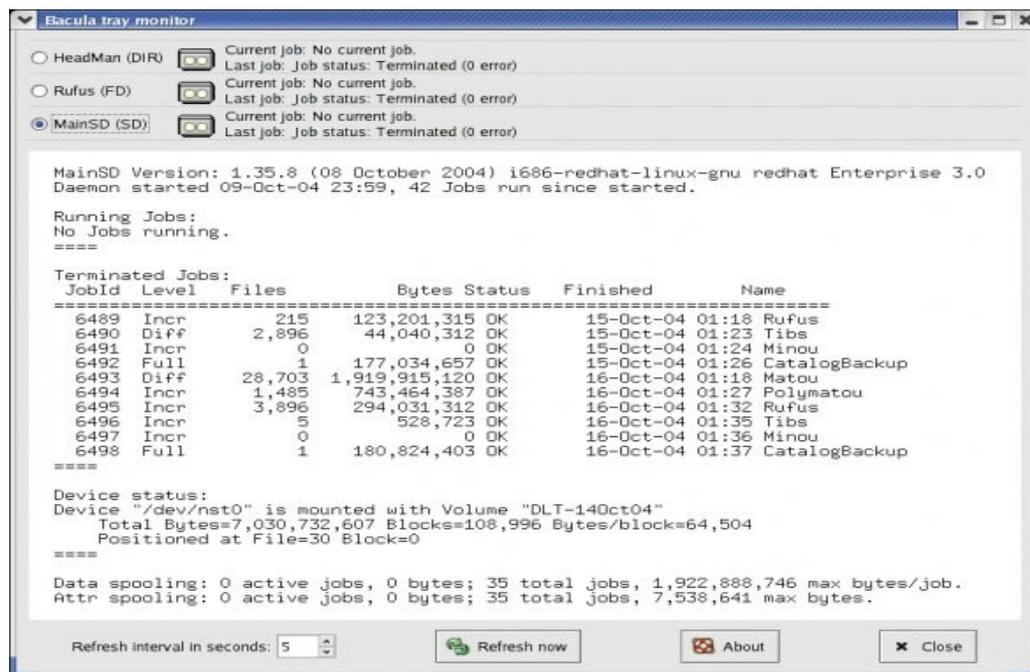
```
*status client
The defined Client resources are:
  1: cliente1-fd
    Connecting to Client cliente1-fd at cliente1.x.com:9102
cliente1-fd Version: 1.32f-5 (09 Mar 2004) i386-pc-linux-gnu debian testing/unstable
Daemon started 30-Apr-04 11:51, 7 Jobs run.
```

```
Terminated Jobs:
JobId Level  Files      Bytes Status  Finished      Name
-----
  40 Sinc   4,463    92,982,884 OK    30-Apr-04 12:03 cliente1-diaria
  52 Sinc    532     3,741,110 OK    30-Apr-04 20:03 cliente1-diaria
  65 Sinc   3,778    75,600,731 OK    03-May-04 20:04 cliente1-diaria
  78 Sinc   2,139    59,748,148 OK    04-May-04 20:04 cliente1-diaria
  90 Full  18,689   175,881,481 OK    05-May-04 13:56 cliente1-semanal
  91 Sinc   2,109    6,092,577 OK    05-May-04 20:03 cliente1-diaria
 114 Sinc   2,347    6,960,272 OK    06-May-04 20:03 cliente1-diaria
```

```
Director connected at: 07-May-04 11:54
No jobs running.
```

Vemos como el cliente funciona y además nos informa de los trabajos realizados hasta la fecha, los archivos copiados el tipo de trabajo realizado y si hay alguno en ejecución. Es el momento de hacer la comprobación para cada uno de los clientes.

El comando *status* admite además los parámetros *dir* y *all*, también lo podemos ver ejecutando *bacula try monitor*



Si queremos información de la situación de los pools, volúmenes, trabajos realizados etc, haremos uso del comando *list* por ejemplo:

```
*list nextvol job=cliente1-diaria
```

The next Volume to be used by Job "cliente1-diaria" will be Diaria1

Nos informa de cual será el volumen requerido para el proximo trabajo de cliente1-diaria

El comando *messages* nos mostrará los mensajes que tenga pendiente el director por comunicarnos aunque también los recibiremos por correo, si así lo tenemos configurado. Otra orden que no podía faltar es *help* la cual nos mostrara el juego completo de ordenes y su misión.

Ejecución de los trabajos.

Aunque la manera normal de funcionar de Báculo, es programar la ejecución de los trabajos mediante un calendario. Es posible ejecutarlos manualmente, para comprobar su correcto funcionamiento. La forma de hacerlo es con el comando *run*

run

```
A job name must be specified.
The defined Job resources are:
1: cliente1-diaria
```

```
Select Job resource (1): 1
Run Backup job
JobName: cliente1-diaria
FileSet: Diaria
Level: Incremental
Client: cliente1-fd
Storage: tape1
Pool: Diaria
When: 2004-05-07 12:33:28
Priority: 10
```

OK to run? (yes/mod/no):

En este momento se nos pregunta si queremos pasar a la ejecución del trabajo, modificar los diferentes parámetros o su cancelación. Es posible alterar los valores que tuviéramos por defecto para ese trabajo si lo hacemos manualmente. Quizá nos interesara copiar los datos en otra parte, o hacer una copia completa etc.

OK to run? (yes/mod/no): yes

Run command submitted.

*

*status client

The defined Client resources are:

1: cliente1-fd

cliente1-fd Version: 1.32f-5 (09 Mar 2004) i386-pc-linux-gnu debian testing/unstable

Daemon started 30-Apr-04 11:51, 7 Jobs run.

Director connected at: 07-May-04 12:37

JobId 126 Job cliente1-diaria.2004-05-07_12.36.48 is running.

Backup Job started: 07-May-04 12:36

Files=10 Bytes=115,580 Bytes/sec=3,611

Files Examined=1,346

Processing file: /var/cache/locate/locatedb

SDReadSeqNo=5 fd=7

Y comprobamos su ejecución mediante *status client*.

Recuperando Archivos

Para recuperar archivos desde las cintas se usa el comando *restore*, aunque previamente deberíamos crear un trabajo que cumpla esa funcionalidad. Puesto que no podemos contemplar a priori que necesitamos restaurar es útil definir un trabajo por defecto y modificar despues cuando nos lo soliciten los valores necesarios, como el cliente, el archivo o directorio que necesitemos, etc. Despues Báculo nos preguntará a cerca del tipo de restauración que queremos llevar a cabo, como vemos la lista es extensa:

```
o select the Joblds, you have the following choices:
  1: List last 20 Jobs run
  2: List Jobs where a given File is saved
  3: Enter list of Joblds to select
  4: Enter SQL list command
  5: Select the most recent backup for a client
  6: Select backup for a client before a specified time
  7: Enter a list of files to restore
  8: Enter a list of files to restore before a specified time
  9: Cancel
Select item: (1-9):
```

Una vez hecha la selección, Báculo nos preguntará por el cliente y acto seguido por el juego de archivos (FileSet) definido. En este momento Báculo realizará una consulta a la base de datos para intentar localizar los trabajos que le hemos pedido, y nos presenta el prompt \$ a la espera de recibir los comandos necesarios para realizar el trabajo de restauración. Si escribimos *help* obtendremos una explicación de cuales son estos comando y su cometido.

Una vez que hemos marcado los archivos o directorio a restaurar, escribimos *done* y Báculo nos pedirá los volúmenes donde estén situados esos archivos. Solicitará confirmación para ejecutar el trabajo presentando las opciones del trabajo que va a llevar a cabo y una vez lo obtenga empezará con el mismo.

Podrá descargar backula desde:

<http://heanet.dl.sourceforge.net/sourceforge/bacula/bacula-2.4.0.tar.gz>

Para profundizar en este tema consulte en

<http://libertonia.escomposlinux.org/story/2004/5/5/93545/27085>

<http://libertonia.escomposlinux.org/story/2004/5/5/1484/20235>

<http://libertonia.escomposlinux.org/story/2004/5/5/1484/20235>

Amanda

Advanced Maryland Automatic Network Disk Archiver ⁴

Archivador Automático Avanzado de Disco en Red de Maryland o Amanda. Se trata de un software de aplicación capaz de realizar copias de seguridad de los datos existentes en una red de computadoras. Desarrollado originalmente por James da Silva durante 1992 para el departamento de informática de la universidad de Maryland, en la actualidad el programa es mantenido por un grupo de voluntarios. Este se distribuye gratuitamente como código abierto bajo una licencia de tipo BSD aunque existen distribuciones comerciales como Amanda Enterprise. Amanda funciona en una gran cantidad de sistemas UNIX, distribuciones de Linux y versiones de Windows, esta última mediante Samba (programa) y/o Cygwin aunque se está programando un cliente nativo.

Funcionamiento

- amandad: Proceso del cliente que ejecuta las peticiones del servidor invocando otros comandos.
- amdump: Comando que inicia el proceso de copia en el servidor basándose en la configuración definida por el usuario
- Medio de almacenamiento: Soporte en el cual queda almacenada la copia

Principales características

- Arquitectura cliente servidor: Permite la posibilidad de realizar copias de sistemas físicamente alejados pero conectados a una misma red y genera un entorno fácilmente escalable.
- Uso de formatos libres: Amanda utiliza herramientas de código abierto como GNUtar o dump.
- Seguridad: Las comunicaciones cliente-servidor pueden ser protegidas usando OpenSSH o algoritmos de cifrado.
- Cache en disco: Amanda almacena las copias en un disco para evitar pérdidas de datos y mejorar la velocidad de grabación.
- Programación adaptable: Se puede utilizar una programación de copias laxa para que Amanda decida el mejor momento para realizarlas aprovechando mejor los recursos del servidor.

⁴ http://es.wikipedia.org/wiki/Advanced_Maryland_Automatic_Network_Disk_Archiver#Principales_caracteristicas

- Gestión de dispositivos: Capacidad para gestionar cambiadores y diferentes soportes de grabación como cintas, discos duros, CD-ROM y otros medios ópticos.

Última versión

La última versión estable es la 2.5.2p1, lanzada el 6 de junio de 2007.

Para qué sirve Amanda ⁵

Se usa para hacer copias de seguridad (backups). Amanda te permite establecer un único servidor de copias de seguridad (tu server Linux) para salvaguardar datos de múltiples máquinas en un mismo dispositivo de copia (también trabaja con un buen número de stackers o apiladores de cintas). Amanda puede usar diferentes programas para realizar las copias, tales como programas de copia comerciales o el simple GNUtar, y puede hacer copias de un gran número de estaciones clientes corriendo múltiples versiones de Unix. Las versiones más recientes de Amanda también pueden usar Samba para hacer copias de máquinas Windows (95/98/NT/2000) en el servidor.

Es decir, Amanda permite salvaguardar de forma automatizada la información importante de tu red, ya esté ubicada en el servidor central, o en los clientes Windows/Unix.

Si tengo problemas ¿A quién acudo?

En caso de problemas son muy recomendables las listas de correo de Amanda. En concreto, suscríbete a <amanda-user@amanda.org>. Date de alta en la lista desde la [página web de Amanda](http://www.amanda.org/). (<http://www.amanda.org/>)

Un ejemplo práctico

Imagina la siguiente situación: eres el administrador de una red de 30 puestos, todos ellos clientes Windows de un servidor Linux. Los clientes Windows almacenan sus documentos importantes en la carpeta "Mis Documentos", y no quieren la responsabilidad de tener que hacer copias de seguridad de su información (dicen que bastante tienen ya con trabajar). Esa red de 30 puestos está servida por una máquina Linux que, entre otras muchas cosas, les da salida a Internet, correo interno/externo, acceso a ficheros de la empresa ubicados en el servidor Linux desde las máquinas windows (a través de Samba), etc. Esa máquina Linux dispone de una unidad de cinta (streamer), con suficiente espacio para almacenar tanto los contenidos del servidor como los de las carpetas "Mis Documentos" de los clientes. Pues bien, con Amanda puedes programar la copia de toda esa información. Además, la puedes automatizar, añadiendo una simple orden en el crontab del servidor.

⁵ http://www.sergio-gonzalez.com/personales/ingenieria_informatica/sistemas_informaticos/documentacion/amanda/amanda.html

1. Obtención de Amanda e Instalación de Paquetes Relacionados:

Existen dos posibilidades distintas a la hora de instalarlo. La primera consiste en la instalación y configuración de forma "manual" de Amanda y todos los paquetes relacionados necesarios para su funcionamiento. Esta opción, si bien requiere un poco más de tiempo y esfuerzo, nos permite un mayor control sobre la configuración en el momento de la instalación (mediante el paso de parámetros a la hora de configurar y compilar).

La segunda opción consiste en la utilización del comando apt-get (como veremos a continuación) que realiza la configuración e instalación de forma automática.

En este caso describiremos ambas instalaciones, si bien es recomendable el uso de la instalación automática, sobre todo a los usuarios más noveles en Linux, ya que configura por sí solo la mayoría de las cosas (siempre será necesario adaptar la configuración a nuestro caso particular).

1.1. Instalación Manual

1.1.1. Obtención de la Última Distribución de Amanda

Algunas de las distribuciones existentes actualmente incorporan Amanda "de serie". A pesar de ello no está de más dirigirnos a la página web de Amanda a fin de comprobar si existe una versión más actual que la que tenemos. Conviene aclarar algunos conceptos antes de elegir la distribución que nos bajaremos.

Amanda viene como distribución de fuentes. Algunos distribuidores de sistemas operativos proporcionan versiones precompiladas de Amanda, pero debido a que Amanda incluye algunos valores en determinados programas, puede que no concuerden con la configuración. Se está trabajando en mover estos valores a archivos de configuración de tiempo de ejecución, pero por ahora Amanda debería ser generado desde código fuente.

La típica distribución de Amanda es un gzip comprimido con un nombre de archivo tal como amanda-2.6.0.tar.gz, lo cual significa que la versión es la 2.6.0. Existen ocasionales parches de las versiones que tienen un nombre como amanda-2.6.0p1.tar.gz (versión 2.6.0 más parche versión 2). Las versiones Beta tienen nombres como amanda-2.6.0b3.tar.gz (tercera beta pre-versión de la versión 2.6.0).

Es altamente recomendable utilizar una versión estable de la distribución y cuyo parche sea el más actual. La última versión estable es amanda-2.6.0p2.tar.gz.

1.1.2. Instalación de Paquetes Relacionados

Esto hay que tenerlo muy en cuenta. Amanda no es independiente de otro software. Otros paquetes pueden ser requeridos para completar nuestra instalación. Antes de continuar, se debería localizar e instalar los paquetes necesarios. Son los siguientes:

- GNU tar 1.12 o superior (<http://www.gnu.org>)

La versión GNU del programa "tar" con capacidades para realizar copias parciales y omitir los archivos seleccionados. Este es uno de los programas clientes de realización de copias que Amanda sabe utilizar.

- Samba 1.9.18p10 o superior (<http://www.samba.org>, y la "Traducción del Manual de Samba", en S.O.B.L.)

Samba es una implementación del protocolo "System Message Block" (SMB) usado por los sistemas basados en Windows para el acceso a archivos. Contiene una herramienta, "smbclient", que Amanda

puede usar para realizar copias a través de Samba.

- Perl 5.004 o superior (<http://www.perl.org>)

Perl es un lenguaje de programación tipo script, orientado a la administración de sistema y la manipulación de textos. Es usado por una serie de herramientas de informes de Amanda y por algunos intercambiadores de cintas.

- GNU readline 2.2.1 o superior (<http://www.gnu.org>)

La librería "GNU readline" puede ser incorporada para su uso por programas interactivos, para proporcionar históricos de línea de comando y para edición. Se crea en la herramienta de restauración de Amanda "amrecover", si está disponible.

- GNU awk 3.0.3 o superior (<http://www.gnu.org>)

La versión GNU del lenguaje de programación "awk" contiene una versión común a plataformas y algunas características adicionales. Es usada por la herramienta opcional de estadísticas de Amanda "amplot".

- gnuplot 3.5 o superior (<ftp://ftp.dartmouth.edu/pub/gnuplot/>)

Esta librería "gnuplot" (que no tiene nada que ver con las herramientas GNU, mira el archivo README de la distribución) es un paquete gráfico de plotado. Se usa por la herramienta de uso opcional de estadísticas de Amanda "amplot".

Debemos tener en cuenta, revisar el directorio de parches de Amanda y mirar en la sección de parches de la página web, para posibles necesidades de actualizaciones de estos paquetes. Las versiones de Samba anteriores a la 2.0.3, en particular, deben ser parcheadas para que funcionen correctamente con Amanda. Sin estos parches, las copias de seguridad parecerán que se están realizando correctamente, pero las imágenes resultantes estarán corruptas.

Cuando Amanda es configurado, las localizaciones de software adicional usado en los clientes, tales como GNU tar y Samba, se incorporan a los programas de Amanda, de forma que el soft adicional debe ser instalado en el mismo sitio donde se encuentra instalado Amanda y en todos los clientes.

1.2. Instalación y Configuración Automáticas con "apt-get"

Como se comentó anteriormente, el comando apt-get permite la instalación de Amanda de forma automática, al igual que resuelve, instala y configura las dependencias que pueda tener ésta con otros paquetes. Para ello tecleamos la siguiente orden:

```
# apt-get install amanda-server amanda-client amanda-common
```

Como se puede ver, estamos indicando que sean instalados los paquetes correspondientes al servidor, cliente y los comunes. Así pues, en el caso de un cliente sólo sería necesario instalar el paquete amanda-client.

2. Compilación e Instalación de Amanda

Este apartado sólo será necesario en caso de haber elegido la instalación manual. Si hemos realizado la instalación mediante el comando apt-get, todo lo que aquí se explica habrá sido ya configurado de forma automática (mediante la configuración por defecto).

Antes de nada debemos descomprimir el paquete (en un directorio cualquiera, por ejemplo

/usr/local/src).

2.1. Configuración Preliminar

La típica configuración de Amanda se ejecuta como un usuario distinto a root, como por ejemplo "backup" o "amanda", con los permisos necesarios para relizar copias de seguridad (en adelante, backups). Frecuentemente, la posibilidad de hacer un login para el usuario que se cree está desactivada (recomendado). Por lo tanto si probamos a hacer un login con ese nombre de usuario, comprobaremos que no puede entrar al sistema.

Para usar el programa comercial "dump" en lugar del GNU tar, el usuario que controla Amanda debe estar en un grupo con permisos de lectura contra los dispositivos de copia. El número de miembros en este grupo debería estar muy controlado, ya que tienen permiso para abrir cada uno de los archivos de los clientes.

Hay dos formas para enlazar Amanda y el grupo de miembros del dispositivo de copia. Poniendo al usuario Amanda en el grupo al que actualmente pertenecen los dispositivos de copia, como el grupo primario o el secundario, o creando un nuevo grupo para Amanda y pasando a este nuevo grupo a los propietarios de los dispositivos. Amanda (actualmente, el programa "dump") necesita acceso de sólo lectura, de modo que desactiva los permisos de escritura del grupo. Desactiva también los permisos del "resto del mundo". Nosotros asociaremos el usuario "amanda" al grupo "disk".

Para usar GNU tar, Amanda funciona bajo un programa setuid-root que garantiza los permisos necesarios. La versión GNU de "tar" debe ser la usada con Amanda. Las versiones comerciales (a menos que tengan origen en la GNU y sean al menos versión 1.12) no funcionarán, porque Amanda depende de características adicionales.

2.2. Configuración de la Generación de Amanda

Ahora viene la parte del ./configure. Este es el típico script con el que nos encontramos en todo paquete de fuentes, que sirve para preparar las órdenes que posteriormente vamos a pasarle al compilador.

Usa el usuario y grupo Amanda para las opciones --with-user y --with-group en ./configure. Por ejemplo, para usar "amanda" como nombre de usuario y "disk" como nombre de grupo:

```
# ./configure --with-user=amanda --with-group=disk
```

No se necesitan obligatoriamente más opciones para ./configure, pero se puede ver todas las posibilidades con ./configure --help, aunque los valores por defecto son normalmente adecuados, y se requiere de cierta experiencia con Amanda para hacer cambios en dichos valores. Dejamos --with-debugging para que los archivos de depuración de errores sean creados en los clientes. Consumirán algo de espacio, pero son "casi" necesarios y sobre todo muy útiles a la hora de resolver posibles problemas.

La generación normal crea tanto el servidor de cintas como el software cliente. La parte servidora del sistema necesita las partes de cliente. Sin embargo, los clientes normalmente no necesitan la parte servidora de Amanda. Te puedes ahorrar tiempo de compilación añadiendo --without-server a los argumentos de ./configure cuando compiles para ellos.

El mecanismo de seguridad por defecto usa un archivo con el mismo formato que .rhosts, pero llamado .amandahosts. Esto mantiene a las operaciones de Amanda separadas del trabajo normal rsh/rcp que usa el mismo usuario. No es recomendable, pero .rhosts y hosts.equiv pueden ser usados, añadiendo

--without-amandahosts a los argumentos de ./configure. Mejor usa .amandahosts.

Amanda usa sus propios protocolos TCP y UDP. Normalmente, los que ves a continuación (extraído del /etc/services):

```
amanda 10080/tcp # Amanda backup services
amanda 10080/udp # Amanda backup services
amandaix 10082/tcp # Amanda backup services
amidxtape 10083/tcp # Amanda backup services
```

Los puertos TCP usados para la transferencia de datos pueden restringirse con --with-portrange para usar Amanda entre hosts separados por un cortafuegos (firewall). Una entrada típica podría ser:

```
# ./configure --with-portrange=50000,50100 ...
```

Esto no afecta a las peticiones UDP iniciales hechas desde el servidor de cintas a los clientes. El puerto UDP de amanda (normalmente 10080) debe tener paso permitido a través del firewall. No debe olvidarse esto, si se usa un cortafuegos.

Si se va a usar muchas opciones con ./configure, se las puede poner todas juntas en /usr/local/share/config.site o en /usr/local/etc/config.site para mantenerlas entre una compilación y otra.

2.3. Compilación e Instalación

Una vez ejecutado ./configure, tecleamos make para compilar Amanda, y luego make install para instalarlo. El paso "make install" se debe hacer necesariamente como root porque algunos programas de Amanda requieren privilegios de sistema.

A menos que se haya cambiado la ubicación por defecto, Amanda se instala en estas áreas:

- /usr/local/sbin: Programas que ejecutan los administradores.
- /usr/local/lib: Librerías.
- /usr/local/libexec: Programas privados que sólo usa Amanda.
- /usr/local/man: Documentación.

Los siguientes programas deben tener setuid-root (lo cual lo hace si tecléa make install como root). El primer grupo (amcheck, dumper, y planner) corren en la máquina servidora de cintas, y necesitan un puerto privilegiado de red para una comunicación segura con los clientes. El resto de programas son utilidades usadas en los clientes, en función del programa de copia usado y el tipo de s.o.

- sbin/amcheck: Programa de chequeo de Amanda.
- libexec/dumper: Programa cliente de comunicaciones.
- libexec/planner: Programa de estimación de porcentaje de volumen copiado.
- libexec/killgrp: Usado para matar (kill) programas de copia que corren como root.
- libexec/rundump: Setuid wrapper para sistemas que necesitan correr el programa de copiado como root.
- libexec/runtar: Setuid wrapper para ejecutar GNU tar como root.

Todos estos programas son instalados con los permisos de acceso de grupo y del "resto del mundo"

desactivados para el grupo de Amanda, denominado según `--with-group`. Asegúrate de que todos los miembros del grupo son los que deben ser, ya que `rundump` y `runtar` en particular dan acceso a cada uno de los archivos en el sistema.

Si el software Amanda está accesible via NFS, hay que asegurarse de que las opciones `mount` admiten programas `setuid`. Además, si se usa GNU `tar`, el `root` necesita acceso de escritura a `/usr/local/var/amanda/gnutar-lists` (o el valor que se haya usado con `--with-gnutar-list` para el `./configure`) para almacenar información sobre cada nivel parcial.

Si la generación te da problemas o Amanda necesita volver a ser generado, sobre todo si se van a agregar nuevas opciones para `./configure`, la siguiente secuencia te garantiza que todo es perfectamente instalado, eliminando cualquier rastro de la instalación anterior:

```
# make distclean
# ./configure ...
# make
# make install (como root)
```

Se pueden diagnosticar posibles problemas en los procesos de `./configure`, mirando en el archivo `config.log`. Contiene una salida detallada de los tests que realiza `./configure`. Advierte que es normal que algunos de dichos tests "fallen", como consecuencia de las pruebas que `./configure` realiza para determinar cómo acceder a varias características del sistema.

Un problema común cuando usa el compilador GNU C es no reinstalarlo cuando cambia la versión de S.O. GCC es particularmente sensible a los archivos de cabecera del sistema y debe ser reinstalado o tener incluidos las modificaciones adecuadas en sus archivos incluye si el S.O. es actualizado. Ejecutando `gcc --verbose` veremos de dónde obtiene `gcc` su información, y contiene una indicación de la versión de s.o. que espera encontrar.

Amanda necesita realizar cambios en los servicios de red y en el archivo de configuración relativos a `inetd`. El script `client-src/patch-system` debería actualizar el sistema en la mayoría de los casos. Este no maneja actualmente sistemas que entreguen servicios via YP/NIS. Si el script no funciona, se debe añadir las siguientes entradas al archivo `services` (p.e., `/etc/services`) or YP/NIS map:

```
Amanda 10080/udp
Amandaidx 10082/tcp
Amidxtape 10083/tcp
```

Suele venir todo preconfigurado. Tan sólo hay que descomentar las líneas en el `inetd.conf`. Pero de todas formas, verifiquemos el `/etc/services`, que nunca está de más.

Cada cliente Unix/Linux necesita una entrada como estas en su archivo de configuración de `inetd` (p.e., `/etc/inetd.conf`), sustituyendo al "usuario_Amanda" por Amanda y la ruta completa al directorio `libexec` de Amanda para RUTA:

```
amanda dgram udp wait usuario_Amanda /RUTA/libexec/amandad amandad
```

El servicio `amanda` es usado por todos los servicios controlados por Amanda para realizar funciones en los clientes. El servidor de cintas necesita entradas como éstas para poder ejecutar la herramienta `amrecover`:

```
amandaidx stream tcp nowait Amanda /PATH/libexec/amindexd amindexd
amidxtape stream tcp nowait Amanda /PATH/libexec/amidxtaped amidxtaped
```

En algunas distribuciones ya existen estas líneas, aunque comentadas. Simplemente hay que descomentarlas, y comprobar que la ruta de acceso a `amindexd` y a `amidxtaped` concuerdan con las de nuestra configuración. Si no es así, las cambiamos. No olvidemos hacer un `killall -HUP inetd` si hemos hecho cambios al `inetd.conf`.

El servicio `amandaidx` proporciona acceso a los catálogos, mientras que el servicio `amidxtape` proporciona acceso remoto al dispositivo de cinta.

3. Pasos Previos a la Puesta en Marcha

Si todo ha ido bien, tenemos ya instalado el software servidor y las herramientas necesarias para hacer copias y restaurarlas.

3.1. Selección del Tipo de Dispositivo de Copia

Si optamos por usar un disco duro como soporte de destino para las copias de seguridad contamos con varias ventajas: permite una mayor velocidad en la realización de las copias, no es necesario estar pendiente de colocar la cinta adecuada, resulta más económico utilizar un dispositivo existente que tener que adquirir uno nuevo, nos evitamos tener una estantería repleta de cintas de copias de seguridad..., etc.

Puesto que Amanda es un servidor de copias de seguridad inicialmente pensado para el almacenamiento de las mismas en cintas será necesario realizar algunas modificaciones en los archivos de configuración inicial, para poder usar un disco duro como dispositivo de copia.

3.2. Selección del Tipo de Copia

Las imágenes de las copias pueden opcionalmente ser comprimidas en el cliente o el servidor de copias (o mediante el hardware del dispositivo de cinta, si es ese el caso). La compresión via software permite a Amanda rastrear el uso y hacer mejores estimaciones de los tamaños de las imágenes, pero la compresión hardware es más eficiente en cuanto al consumo de recursos de la CPU. Desactivaremos la compresión de hardware cuando uses compresión software en el cliente o el servidor. Estudiaremos la documentación del S.O. para ver cómo se controla la compresión via hardware; en muchos sistemas esto se hace a través del nombre del archivo del dispositivo con el flag `no-rebobinable`. AIX usa el comando `chdev`.

4. Configuración de Amanda

En esta parte se realiza la configuración del funcionamiento de Amanda. Básicamente, tenemos que tocar dos archivos. El que le dice a Amanda cómo debe funcionar, `amanda.conf`, y el que le indica qué es lo que tiene que copiar, `disklist`.

4.1. El archivo de Configuración "amanda.conf"

Elegimos un nombre para la configuración, por ejemplo "Diaria". Creamos un directorio en la máquina servidora de cintas para almacenar los archivos de configuración, normalmente en `/usr/local/etc/amanda/Diaria`. Accedemos al nuevo directorio. Su uso debería estar restringido al grupo de Amanda, o mejor, para acceso sólo del usuario Amanda.

Amanda limita el uso de red, de forma que las copias de seguridad no acaparen toda la capacidad del sistema. Este límite es impuesto cuando Amanda está decidiendo si realizar una copia estimando la salida y añadiéndola a las copias que se están ejecutando en ese momento. Si el valor excede del ancho de banda

asignado a Amanda, entonces la copia es retenida hasta que las otras hayan terminado. Una vez que se inicia una copia, Amanda permite a otros componentes de la red ejecutar cualquier operación, reduciendo su predominio.

Copiamos la plantilla de ejemplo del archivo `example/amanda.conf` al directorio de configuración que hemos creado y lo editamos. Encontraremos una completa información sobre su contenido en la página man de Amanda. Comenzamos con los siguientes (algunos de ellos se explicarán más adelante):

- `org`: Esta cadena estará en la línea de Asunto (Subject) de los reportes recibidos vía e-mail desde Amanda.
- `mailto`: Usuarios destinatarios de los reportes que genera Amanda y envía vía e-mail.
- `dumpuser`: Igual que `--with-user` en `./configure`.
- `dumpcycle`: Ciclo de Copia.
- `runspcycle`: Ejecuciones por ciclo.
- `tapecycle`: Número mínimo de cintas necesarias para el ciclo.
- `runtapes`: Número de cintas a usar por ejecución.
- `tapedev`: El dispositivo de cinta "no-rewind" (no rebobinable) si no se va a usar un cambiador de cintas, o si se va a usar el cambio manual.
- `tapetype`: Tipo de cinta.
- `netusage`: Ancho de banda de la Red asignado a Amanda.
- `labelstr`: Una expresión regular (grep pattern) usada para garantizar que cada cinta está asignada a esa configuración de Amanda. Nuestro ejemplo debería usar "Diaria-[0-9][0-9][0-9]".

Los siguientes parámetros probablemente no necesitarán ser cambiados.

- `infofile`: Localización de la B.D. que contiene el histórico de operaciones de Amanda. Versiones antiguas de Amanda usan esto como el nombre base de un archivo de bases de datos. Las nuevas versiones lo usan como nombre de directorio.
- `logdir`: Directorio donde los registros de Amanda son almacenados.
- `indexdir`: Localización de la base de datos de catálogos (opcional) de Amanda.

Es necesario realizar las siguientes modificaciones en el presente archivo de configuración, a fin de seleccionar el disco duro como dispositivo de almacenamiento de las copias de seguridad. Esta configuración es genérica para el uso de cualquier tipo de disco duro:

1. Ponemos el parámetro `tapedevice` a "no-such-device",
2. Ponemos `rawtapedev` a "no-such-device",
3. Ponemos `changerdev` a "no-such-device",
4. Ponemos `tapetype` a `DISKSAVE`,
5. Definimos un nuevo tipo de cinta, que en este caso representa al disco duro, y que llamaremos `DISKSAVE` por ejemplo. Para ello añadimos las siguientes líneas en la sección `tapetypes` del archivo:
define
tapetype `DISKSAVE` {
 comment "Fake tape description for save to disk"
 length 1000 gbytes
 filemark 0 kbytes
 speed 2000 kbytes
}
6. Comentamos la sección en la que se define el "holdingdisk".

7. Ponemos reserve a 30 (por ejemplo).
8. Comentamos el parámetro runtapes.

4.2. El archivo de Configuración "disklist"

Una vez que el archivo amanda.conf ha sido correctamente configurado, escogemos al primer cliente, normalmente el propio servidor, y los sistemas de archivos o directorios a copiar. Por cada área a salvaguardar, seleccionamos el programa de copia que usaremos (GNU tar). Los programas de copia comerciales suelen ser más eficientes y no perturban a los archivos que están copiando, pero normalmente no son portables entre diferentes sistemas operativos. GNU tar es portable y tiene algunas características adicionales, como la habilidad de excluir patrones de archivos, pero altera el último tiempo de acceso para cada archivo copiado y puede no llegar a ser tan eficiente. GNU tar también puede repartirse con sistemas de archivos activos mejor que los programas de copia comerciales, y es capaz de manejar sistemas de archivos muy grandes, rompiéndolos en subdirectorios.

A continuación seleccionaremos el tipo de compresión para cada área, si la hay. Consideramos desactivar la compresión de las áreas críticas, necesarias para recuperar el sistema de una máquina, en el caso de que el programa de descompresión no esté disponible. La compresión de cliente extiende la carga a múltiples máquinas y reduce el tráfico de la red, pero puede no resultar apropiada para el caso de clientes lentos o bien ocupados con mucha carga de procesos. La compresión de Servidor incrementa la carga del servidor de cintas. En su lugar, si usamos GNU gzip, la compresión puede hacerse de forma más rápida y menos agresiva. Establecemos la compresión a ninguna para desactivar compresión via software o usar compresión via hardware.

Escoge o modifica un tipo de copia o dumptype ya existente que coincida con las opciones que desees, o crea uno nuevo. Cada dumptype debería referenciar al dumptype global. Este es usado para establecer opciones para el resto de dumptypes. Por ejemplo, para usar la característica de indexación o indexing, actívala en el dumptype global, y los demás tipos que definas heredarán ese valor. Para nuestra instalación, y por los motivos antes comentados hemos seleccionado siempre aquellos dumptypes que hacen uso del programa GNU tar.

La capacidad de indexación genera un catálogo comprimido de cada imagen de copia. Esto es útil para encontrar archivos perdidos, y es la base del programa amrecover. Ciclos de copia muy grandes o áreas con muchos o muy activos archivos pueden provocar que los catálogos usen mucho espacio en disco. Amanda automáticamente elimina los catálogos de las imágenes que ya no están en el disco (u otro dispositivo de almacenamiento usado).

Creemos un archivo llamado disklist en el mismo directorio donde reside tu amanda.conf o bien copia el que tienes en example/disklist. Nos aseguramos de que es legible por el usuario Amanda. Cada línea en disklist define un área a ser copiada. El primer campo es el nombre de la máquina cliente (se aconsejan nombres completamente cualificados de dominio), el segundo es el área a ser salvaguardada en el cliente, y el tercero es el método de copia, o dumptype. El área puede introducirse como nombre de disco, sd0a, como nombre de dispositivo, /dev/rsd0a, o como nombre lógico, /usr. Los nombres lógicos son más fáciles para recordar qué es lo que se está copiando, así como a la hora de restauración o la reconfiguración del disco.

Para configurar un cliente Windows, estableceremos el nombre de la máquina al nombre de la máquina Unix que corre Samba (el mismo servidor Linux que corre Amanda es nuestro caso) y el área al nombre del recurso compartido de Windows, como por ejemplo //algun-pc/C\$. Las barras que se usan como separadores son las de Unix, y no las de Windows.

Activamos el acceso de Amanda al cliente desde el servidor de cintas (a menos que el cliente sea el propio servidor de cintas) editando el archivo `.amandahosts` (o `.rhosts`, dependiendo de la configuración que realizamos en `./configure`) en el directorio raíz del usuario Amanda en el cliente. Introducimos el nombre completamente cualificado de dominio del servidor de copias de seguridad y el usuario Amanda (o el correspondiente en cada caso), separados por un espacio o tabulador. Nos aseguramos de que el archivo es propiedad del usuario Amanda y no permite acceso a nadie más que al propietario (p.e. modo 0600 o 0400).

Para los clientes Windows, colocamos la contraseña del recurso en `/etc/amandapass` en el servidor que corre Samba. El primer campo es el nombre de recurso compartido de Windows, el segundo es la contraseña en modo texto, y el tercer campo (opcional) es el dominio. Debido a que este archivo contiene contraseñas visibles, debería estar muy protegido, ser propiedad del usuario Amanda y sólo accesible a él. Por defecto, Amanda usa al usuario Samba. Esto lo puedes cambiar con `--with-samba-user` en `./configure`.

5. Ejemplos de archivos de Configuración

A continuación listamos el resultado de nuestros archivos de configuración: "amanda.conf" y "disklist".

5.1. "amanda.conf"

```
#
# amanda.conf - sample Amanda configuration file.
#
# If your configuration is called, say, "DailySet1", then this file
# normally goes in /etc/amanda/DailySet1/amanda.conf.
#
# for explanation of the parameters refer to amanda(8) and
# /usr/doc/amanda/WHATS.NEW.gz

org "Diaria"      # your organization name for reports
mailto "amanda"  # space separated list of operators at your site
dumpuser "amanda" # the user to run dumps under
#
inparallel 4     # maximum dumpers that will run in parallel
netusage 600    # maximum net bandwidth for Amanda, in KB per sec

# a filesystem is due for a full backup once every <dumpcycle> days
dumpcycle 4 weeks # the number of days in the normal dump cycle
tapecycle 8 tapes # the number of tapes in rotation

bumpsize 1 MB   # minimum savings (threshold) to bump level 1 -> 2
bumpdays 1     # minimum days at each level
bumpmult 4     # threshold = bumpsize * (level-1)**bumpmult

#runtapes 9     # explained in WHATS.NEW
#tpchanger "no-changer" # the tape-changer glue script, see TAPE.CHANGERS
tapedev "no-such-device" # Linux @ tuck, important: norewinding
rawtapedev "no-such-device" # the raw device to be used (ftape only)
#changerfile "/mnt/amanda/changer"
changerdev "no-such-device"

tapetype DISKSAVE # what kind of tape it is (see tapetypes below)
labelstr "^HISS[0-9][0-9]*$" # label constraint regex: all tapes must match

diskdir "/mnt/backup" # where the holding disk is
disksize 10 MB       # how much space can we use on it
reserve 30
#diskdir "/dumps/amanda/work" # additionally holding disks can be specified
#diskdir "/mnt/disk4"
#disksize 1000 MB     # they are used round-robin

# Amanda needs a few MB of disk space for the log and debug files,
# as well as a database. This stuff can grow large, so the conf directory
# isn't usually appropriate.
```

```
infile "/var/lib/amanda/DailySet1/curinfo" # database filename
logfile "/var/log/amanda/DailySet1/log" # log filename

# where the index files live
indexdir "/var/lib/amanda/DailySet1/index"

# Specify holding disks. These are used as a temporary staging area for
# dumps before they are written to tape and are recommended for most sites.
# The advantages include: tape drive is more likely to operate in streaming
# mode (which reduces tape and drive wear, reduces total dump time); multiple
# dumps can be done in parallel (which can dramatically reduce total dump time.
# The main disadvantage is that dumps on the holding disk need to be flushed
# (with amflush) to tape after an operating system crash or a tape failure.
# If no holding disks are specified then all dumps will be written directly
# to tape. If a dump is too big to fit on the holding disk than it will be
# written directly to tape. If more than one holding disk is specified then
# they will all be used round-robin.

#holdingdisk hd1 {
# comment "main holding disk"
# directory "/mnt/amanda1" # where the holding disk is
# use 30 Mb # how much space can we use on it
# # a non-positive value means:
# # use all space but that value
# chunksize 1Mb # size of chunk if you want big dump to be
# # dumped on multiple files on holding disks
# # N Kb/Mb/Gb split images in chunks of size N
# # The maximum value should be
# # (MAX_FILE_SIZE - 1Mb)
# # 0 same as INT_MAX bytes
#}

# tapetypes
#
# Define the type of tape you use here, and use it in "tapetype" above.
# Some typical types of tapes are included here. The tapetype tells amanda
# how many MB will fit on the tape, how big the filemarks are, and how
# fast the tape device is.
#
# For completeness Amanda should calculate the inter-record gaps too, but it
# doesn't. For EXABYTE and DAT tapes this is ok. Anyone using 9 tracks for
# amanda and need IRG calculations? Drop me a note if so.

define tapetype DISKSAVE {
comment "Fake tape description for save to disk"
length 1000 gbytes
filemark 0 kbytes
speed 2000 kbytes
}

define tapetype QIC-60 {
comment "Archive Viper"
length 60 mbytes
filemark 100 kbytes # don't know a better value
speed 100 kbytes # dito
}

define tapetype DEC-DLT2000 {
comment "DEC Differential Digital Linear Tape 2000"
length 15000 mbytes
filemark 8 kbytes
speed 1250 kbytes
}

# goluboff@butch.Colorado.EDU
# in amanda-users (Thu Dec 26 01:55:38 MEZ 1996)
define tapetype DLT {
comment "DLT tape drives"
length 20000 mbytes # 20 Gig tapes
filemark 2000 kbytes # I don't know what this means
speed 1500 kbytes
}
```

```
define tapetype SURESTORE-1200E {
    comment "HP AutoLoader"
    length 3900 mbytes
    filemark 100 kbytes
    speed 500 kbytes
}

define tapetype EXB-8500 {
    comment "Exabyte EXB-8500 drive on decent machine"
    length 4200 mbytes
    filemark 48 kbytes
    speed 474 kbytes
}

define tapetype EXB-8200 {
    comment "Exabyte EXB-8200 drive on decent machine"
    length 2200 mbytes
    filemark 2130 kbytes
    speed 240 kbytes
}

define tapetype HP-DAT {
    comment "DAT tape drives"
    length 1900 mbytes      # these numbers are not accurate
    filemark 100 kbytes    # but you get the idea
    speed 500 kbytes
}

define tapetype DAT {
    comment "DAT tape drives"
    length 1000 mbytes     # these numbers are not accurate
    filemark 100 kbytes   # but you get the idea
    speed 100 kbytes
}

define tapetype MIMSY-MEGATAPE {
    comment "Megatape (Exabyte based) drive through Emulex on Vax 8600"
    length 2200 mbytes
    filemark 2130 kbytes
    speed 170 kbytes      # limited by the Emulex bus interface, ugh
}

define tapetype QIC-3080 {
    comment "QIC 3080"
    length 2000 mbytes
    filemark 64 kbytes
    speed 250 kbytes
}

# dumptypes
#
# These are referred to by the disklist file. The dumptype specifies
# certain "options" for dumping including:
#   index          - keep an index of the files backed up
#   compress-fast  - (default) compress on the client using fast algorithm
#   compress-best  - compress using the best (and slowww) algorithm
#   no-compress    - don't compress the dump output
#   srvcompress    - Compress dumps on the tape host instead of client
#                   machines. This may be useful when a fast tape host
#                   is backing up slow clients.
#   record         - (default) record the dump in /etc/dumpdates
#   no-record      - don't record the dump, for testing
#   no-hold        - don't go to the holding disk, good for dumping
#                   the holding disk partition itself.
#   skip-full      - Skip the disk when a level 0 is due, to allow
#                   full backups outside Amanda, eg when the machine
#                   is in single-user mode.
#   skip-incr      - Skip the disk when the level 0 is NOT due. This
#                   is used in archive configurations, where only full
#                   dumps are done and the tapes saved.
#   no-full        - Do a level 1 every night. This can be used, for
```

```
#          example, for small root filesystems that only change
#          slightly relative to a site-wide prototype. Amanda
#          then backs up just the changes.
#
# Also, the dumptype specifies the priority level, where "low", "medium" and
# "high" are the allowed levels. These are only really used when Amanda has
# no tape to write to because of some error. In that "degraded mode", as
# many incrementals as will fit on the holding disk are done, higher priority
# first, to insure the important disks are dumped first.

define dumptype always-full {
    comment "Full dump of this filesystem always"
    options no-compress
    priority high
    dumpcycle 0
    maxcycle 0
}

define dumptype comp-user-tar {
    program "GNUTAR"
    comment "partitions dumped with tar"
    options compress-fast, index, exclude-list "/etc/amanda/exclude.gtar"
    priority medium
}

define dumptype comp-root-tar {
    program "GNUTAR"
    comment "Root partitions with compression"
    options compress-fast, index, exclude-list "/etc/amanda/exclude.gtar"
    priority low
}

define dumptype user-tar {
    program "GNUTAR"
    comment "partitions dumped with tar"
    options no-compress, index, exclude-list "/etc/amanda/exclude.gtar"
    priority medium
}

define dumptype high-tar {
    program "GNUTAR"
    comment "partitions dumped with tar"
    options no-compress, index, exclude-list "/etc/amanda/exclude.gtar"
    priority high
}

define dumptype root-tar {
    program "GNUTAR"
    comment "Root partitions dumped with tar"
    options no-compress, index, exclude-list "/etc/amanda/exclude.gtar"
    priority low
}

define dumptype comp-user {
    comment "Non-root partitions on reasonably fast machines"
    options compress-fast
    priority medium
}

define dumptype nocomp-user {
    comment "Non-root partitions on slow machines"
    options no-compress
    priority medium
}

define dumptype holding-disk {
    comment "The master-host holding disk itself"
    options no-hold
    priority medium
}

define dumptype comp-root {
```

```
    comment "Root partitions with compression"
    options compress-fast
    priority low
}

define dumptype nocomp-root {
    comment "Root partitions without compression"
    options no-compress
    priority low
}

define dumptype comp-high {
    comment "very important partitions on fast machines"
    options compress-best
    priority high
}

define dumptype nocomp-high {
    comment "very important partitions on slow machines"
    options no-compress
    priority high
}

define dumptype nocomp-test {
    comment "test dump without compression, no /etc/dumpdates recording"
    options no-compress, no-record
    priority medium
}

define dumptype comp-test {
    comment "test dump with compression, no /etc/dumpdates recording"
    options compress-fast, no-record
    priority medium
}
```

5.2. "disklist"

```
# sample Amanda2 disklist file, derived from CS.UMD.EDU's disklist
#
# If your configuration is called, say, "DailySet1", then this file
# normally goes in /etc/amanda/DailySet1/disklist.
#
# File format is:
#
#     hostname diskdev dumptype
#
# where the dumptypes are defined by you in amanda.conf.

# Configuración en Lítio

localhost /mnt/copia comp-root-tar
```

6. Ejecutando Amanda

Ya tenemos todo listo para que el sistema funcione. Veremos qué hay que hacer para que Amanda entre en ejecución y nos resuelva el problema de la realización de las copias de seguridad en nuestro sistema.

6.1. Comprobación del Funcionamiento: "amcheck"

Una de las cosas buenas que tiene Amanda es que podemos saber si va a funcionar o no, antes de ejecutarlo. Podemos testear la configuración con el comando `amcheck`. Ejecutamos este comando como usuario `amanda`, y no como `root`:

Si en éste momento estamos como `root`, usamos:

```
# su amanda -c "amcheck Diaria"
```

Si estamos como usuario `amanda`, usamos:

amcheck Diaria

Muchos de los errores reportados por amcheck están descritos en docs/FAQ, o en la página man de amcheck. El error más común reportado es "selfcheck request timed out", lo que significa que amcheck no ha podido hablar con el daemon amandad (miramos el /etc/inetd.conf) en el cliente. Es importante recordar en este punto que las listas de correo funcionan de forma muy eficiente, ya que la gente responde en el mismo día.

Para un testeo inicial, establecemos la opción record a no en el dumptype global, pero recordamos ponerlo a yes cuando Amanda vaya a realizar su funcionamiento normal. Este parámetro controla si el programa de copia en el cliente actualiza su propia base de datos, tal como /etc/dumpdates.

Para empezar completamente desde cero, eliminamos los archivos o directorios nominados en los directorios info e index, el archivo tapelist, y todos los archivos amdump.* en el directorio de configuración, así como todos los archivos del tipo log.*. Estos archivos contienen información histórica que Amanda necesita entre sus ejecuciones y también son necesarios para encontrar imágenes de copias para su restauración. Deberían estar protegidos cuando Amanda entre en funcionamiento en nuestro sistema. Las rutas y nombres posibles de todos estos archivos vienen definidas en el archivo de configuración amanda.conf.

6.2. Ejecución de la Copia: "amdump"

El script amdump controla una ejecución normal de Amanda. Simplemente hay que tener en cuenta estos puntos:

Su ejecución es como sigue: amdump, y a continuación el nombre del tipo de configuración. En nuestro caso, sería: amdump Diaria. Se ejecuta como usuario amanda, no como root. Lo podemos ejecutar manualmente, desde consola, o meterlo como tarea del cron del usuario amanda, para que las copias se hagan automáticamente.

Si cuando se va a ejecutar amdump, este detecta otra copia en ejecución, o una copia anterior abortada, amdump genera un error y se detiene. Usa amcleanup para limpiar el sistema de ejecuciones abortadas, y luego podrás ejecutar de nuevo amdump.

Cuando Amanda termina una ejecución, manda un reporte en forma de correo electrónico al destinatario o destinatarios establecidos en el archivo amanda.conf (recordamos el parámetro "mailto", en las primeras líneas del archivo de configuración). También renombra el archivo de registro de la ejecución a un nombre de archivo único, con el formato log.YYYYMMDD.N nombre.

Es recomendable, al menos durante los primeros días de funcionamiento del sistema, leer los reportes que llegan al correo.

6.3. Recuperación de Datos: "amrecover" y "amrestore"

Es el momento de hablar sobre cómo podemos restaurar la información desde una copia de seguridad a su origen. De eso se encargan los comandos amrecover y amrestore, que tendremos que ejecutar como root. A continuación se explica el uso de ambos, así como las particularidades, ventajas y desventajas de cada uno de ellos.

6.3.1. Configuración y uso de "amrecover"

amrecover es el comando que vamos a usar para restaurar archivos con Amanda. Antes de que amrecover pueda funcionar, Amanda debe ejecutarse con el parámetro dumptype index establecido a yes, y los servicios amindexd y amidxtaped deben estar instalados y activados en el archivo /etc/inetd.conf del servidor de cintas (esto lo hace por defecto la instalación). Además, el cliente debe figurar en el archivo .amandahosts (o .rhosts) del servidor de cintas. Es imprescindible ejecutar el comando amrecover como root. amrecover no debería hacerse setuid-root, ya que podría entonces abrir catálogos del sistema entero para todo el mundo.

Veremos a continuación un ejemplo práctico que ilustra la situación. El usuario X ha pedido que se le restauren dos archivos, ambos con el mismo nombre "molecule.dat", en los subdirectorios llamados work/sample-21 y work/sample-22 y dice que quiere las versiones que se modificaron por última vez el 13 de enero. Nos hacemos root en el cliente donde vamos a restaurar la información (si es un cliente unix/linux. Si no, desde el server), vamos al área y ejecutamos amrecover:

```
$ su
Password:
# cd ~jj
# amrecover Diaria
AMRECOVER Version 2.4.1p1. Contacting server on amanda.cc.purdue.edu ...
220 amanda Amanda index server (2.4.1p1) ready.
200 Access OK
Setting restore date to today (1999-01-18)
200 Working date set to 1999-01-18.
200 Config set to Diaria.
200 Dump host set to pete.cc.purdue.edu.
$CWD '/home/pete/u66/jj' is on disk '/home/pete/u66' mounted at '/home/pete/u66'.
200 Disk set to /home/pete/u66.
amrecover>
```

En este punto, un interfaz de línea de comandos nos permite navegar por la imagen de los catálogos. Navega mediante el comando cd, está disponible con ls, cambia las fechas con setdate, añade archivos y directorios a la lista de extracción con add. El comando extract inicia la extracción de lo que hayamos seleccionado. Y para consultar otros comandos, siempre está el comando help:

```
amrecover> setdate ---14
200 Working date set to 1999-01-14.
amrecover> cd work/sample-21
/home/pete/u66/jj/work/sample-21
amrecover> add molecule.dat
Added /jj/work/sample-21/molecule.dat
amrecover> cd ../sample-22
/home/pete/u66/jj/work/sample-22
amrecover> add molecule.dat
Added /jj/work/sample-22/molecule.dat
amrecover> extract
Extracting files using tape drive /dev/rmt/0mn on host amanda.cc.purdue.edu.
The following tapes are needed: Diaria-034
Restoring files into directory /home/pete/u66
Continue? [Y/n]: y
Load tape Diaria-034 now
Continue? [Y/n]: y
Warning: ./jj: File exists
Warning: ./work: File exists
Warning: ./work/sample-21: File exists
Warning: ./work/sample-22: File exists
set owner/mode for '.'? [yn] n
amrecover> quit
```

amrecover localiza los directorios y archivos que contienen las imágenes, opcionalmente la descomprime (si hemos usado compresión), llega al cliente a través de la red y se entuba en el apropiado

programa de restauración con los argumentos necesarios para extraer los archivos solicitados. amrecover no sabe cómo ejecuta cada cliente el programa de restauración. Mira la página man de amrecover para información actualizada. amrecover no debería usarse para una restauración total del sistema de archivos con herramientas de restauración comerciales, pero funciona de maravilla con el GNU tar. Las herramientas comerciales deberían ejecutarse con el flag r para una restauración completa, y con el flag x para extraer archivos individuales. Una restauración completa con herramientas comerciales debería realizarse con el comando amrestore.

6.3.2. Uso de "amrestore"

Respecto a este comando, comentaremos que su función es recuperar imágenes completas desde el disco o cualquier otro soporte utilizado, aunque está más orientado a dar un soporte para la realización de copias de seguridad en cintas, ya que proporciona facilidades como la búsqueda de cintas mediante indexación de sus contenidos, y otras.

7. Interpretación de los Reportes

Cada vez que ejecutemos el comando amdump para la realización de una copia de seguridad Amanda generará un reporte que nos será directamente enviado al correo del usuario especificado en el archivo de configuración amanda.conf. Este reporte tiene varias secciones que paso a explicar a continuación mediante un ejemplo. Este ejemplo hace referencia al uso de cintas como soporte para las copias, pero la mayoría de las cosas son aplicables al caso del uso de un disco duro como soporte.

```
These dumps were to tape Daily-009, Daily-010
Tonight's dumps should go onto 2 tapes: Daily-011, Daily-012.
```

Estas líneas muestran qué cintas fueron usadas durante la ejecución, y cuáles serán usadas en la próxima vez. En caso de usar un disco duro como soporte harán referencia a éste usado, ya que es este el que almacena dichas copias.

```
FAILURE AND STRANGE DUMP SUMMARY:
gurgi.cc.p /var lev 0 FAILED [Request to gurgi.cc.purdue.edu timed out.]
gurgi.cc.p / lev 0 FAILED [Request to gurgi.cc.purdue.edu timed out.]
pete.cc.pu /var/mail lev 0 FAILED ["data write: Broken pipe"]
samba.cc.p //nt-test.cc.purdue.edu/F$ lev 1 STRANGE
mace.cc.pu /master lev 0 FAILED [dumps too big, but cannot incremental dump new disk]
```

Los problemas que se encontraron durante la ejecución son resumidos en ésta sección. En el ejemplo:

gurgi.cc.purdue.edu estaba caído (apagado), así que todas sus copias fallaron.

El problema en /var/mail en la máquina pete.cc.purdue.edu y el problema F\$ en nt-test.cc.purdue.edu se detallan abajo.

EL área /master en mace.cc.purdue.edu es nueva para Amanda así que se requiere una copia completa, pero esta no cabía en el espacio que le quedaba libre a la cinta.

STATISTICS:

```
Total Full
Daily
```

```
-----
Dump Time (hrs:min)
5:03
3:23
0:33
```

(0:14 start, 0:53 idle)

Output Size (meg)

20434.4
17960.0
2474.4

Original Size (meg)

20434.4
17960.0
2474.4

Avg Compressed Size (%)

Tape Used (%)

137.4
120.0
17.4
(level:#disks ...)

Filesystems Dumped

90
21
69
(1:64 2:2 3:3)

Avg Dump Rate (k/s)

1036.5
1304.3
416.2

Avg Tp Write Rate (k/s)

1477.6
1511.2
1271.9

Esto resume toda la ejecución. Tomó unas cinco horas, unas 3.5 horas escribiendo copias completas y una hora y media para copias parciales. Se tomó 14 minutos para iniciar la ejecución, y la cinta estuvo una hora esperando a las copias que le venían desde el disco de almacenamiento.

En este ejemplo, la compresión hardware fue usada, de forma que Avg Compressed Size no es aplicable y el tamaño de la salida o Output Size escrita a la cinta coincide con el tamaño original de los clientes. Aproximadamente un 137% de la longitud de la cinta tal como se definió en el tapetype fue usado, 120% para copias completas y 17% para parciales. Las líneas Rate nos dan la velocidad de la copia desde el cliente al servidor de cintas y la velocidad de escritura en cinta, todo ello en KBytes por segundo. La línea Filesystems Dumped indica que 90 áreas fueron procesadas, 21 copias completas y 69 parciales. De las parciales, 64 fueron de nivel 1, dos de nivel 2 y tres de nivel 3.

FAILED AND STRANGE DUMP DETAILS:

```
/-- pete.cc.pu /var/mail lev 0 FAILED ["data write: Broken pipe"]
sendbackup: start [pete.cc.purdue.edu:/var/mail level 0]
sendbackup: info BACKUP=/usr/sbin/ufsdump
sendbackup: info RECOVER_CMD=/usr/sbin/ufsrestore -f... -
sendbackup: info end
| DUMP: Writing 32 Kilobyte records
| DUMP: Date of this level 0 dump: Sat Jan 02 02:03:22 1999
| DUMP: Date of last level 0 dump: the epoch
| DUMP: Dumping /dev/md/rdsk/d5 (pete.cc.purdue.edu:/var/mail) to standard output.
| DUMP: Mapping (Pass I) [regular files]
| DUMP: Mapping (Pass II) [directories]
| DUMP: Estimated 13057170 blocks (6375.57MB) on 0.09 tapes.
| DUMP: Dumping (Pass III) [directories]
| DUMP: Dumping (Pass IV) [regular files]
| DUMP: 13.99% done, finished in 1:02
| DUMP: 27.82% done, finished in 0:52
| DUMP: 41.22% done, finished in 0:42
```

```
!-- samba.cc.p //nt-test.cc.purdue.edu/F$ lev 1 STRANGE
sendbackup: start [samba.cc.purdue.edu://nt-test/F$ level 1]
sendbackup: info BACKUP=/usr/local/bin/smbclient
sendbackup: info RECOVER_CMD=/usr/local/bin/smbclient -f... -
sendbackup: info end
- Can't load /usr/local/samba-2.0.2/lib/smb.conf - run testparm to debug it
| session request to NT-TEST.CC.PURD failed
|   directory \top\
|   directory \top\Division\
|   238 ( 2.7 kb/s) \top\Division\contract.txt
|   19456 ( 169.6 kb/s) \top\Division\stuff.doc
...

```

Los fallos y resultados inesperados son detallados aquí. La copia de /var/mail no cambía en la primera cinta, así que la copia fue abortada y vuelta a iniciar en la segunda cinta, tal como se describe en la siguiente sección.

La copia de F\$ en nt-test.cc.purdue.edu falló debido a un problema con la configuración del archivo de configuración de SAMBA. Se marcó como STRANGE porque la línea con una barra "|" no coincide con ninguna de las expresiones regulares construidas en Amanda. Cuando se hacen copias de clientes Windows a través de SAMBA, es normal obtener errores sobre archivos ocupados, tales como PAGEFILE.SYS y el registro. Se deberían hacer otros arreglos para que este tipo de archivos se salvaguardasen correctamente, tales como tareas periódicas en el PC que creasen una copia que no estuviese ocupada en el momento de la ejecución de Amanda.

NOTES:

```
planner: Adding new disk j.cc.purdue.edu:/var.
planner: Adding new disk mace.cc.purdue.edu:/master.
planner: Last full dump of mace.cc.purdue.edu:/src on tape Daily-012 overwritten in 2 runs.
planner: Full dump of loader.cc.purdue.edu:/var promoted from 2 days ahead.
planner: Incremental of sage.cc.purdue.edu:/var bumped to level 2.
taper: tape Daily-009 kb 19567680 fm 90 writing file: short write
taper: retrying pete.cc.purdue.edu:/var/mail.0 on new tape: [writing file: short write]
driver: pete.cc.purdue.edu /var/mail 0 [dump to tape failed, will try again]
taper: tape Daily-010 kb 6201216 fm 1 [OK]

```

Notas informativas sobre la ejecución se listan aquí. Los mensajes dicen:

Hay nuevas entradas en el disklist para j.cc.purdue.edu y para mace.cc.purdue.edu.

La cinta Daily-012 va a ser sobrescrita en dos ejecuciones más y contienen la copia completa más reciente de /src desde mace.cc.purdue.edu, así que el ciclo de la cinta no debería ser más largo.

La siguiente copia completa programada de /var en loader.cc.purdue.edu fue movida dos días para mejorar el balance de la carga.

La copia parcial de /var en sage.cc.purdue.edu se pasó del nivel 1 al nivel 2 debido a que el nivel más alto se estimó que podría tener suficiente espacio para la copia.

El resto de notas dicen que la unidad de cinta no podía escribir todos los datos deseados, probablemente debido a que se llegó al final de la cinta. Llegados a este punto, se habían escrito 19567680 KBytes en 90 archivos sobre la cinta Daily-009. Otro intento de realizar una copia completa de /var/mail desde pete.cc.purdue.edu se hizo en la siguiente cinta (Daily-010) y tuvo éxito, escribiendo 6201216 KBytes en un archivo.

DUMP SUMMARY:

```
DUMPER STATS TAPER STATS
-----
```

```
HOSTNAME DISK L ORIG-KB OUT-KB COMP% MMM:SS KB/s MMM:SS KB/s
boiler.cc / 1 2624 2624 -- 0:13 200.1 0:02 1076.0
boiler.cc /home/boiler/a 1 192 192 -- 0:07 26.7 0:02 118.5
boiler.cc /usr 1 992 992 -- 0:41 24.2 0:02 514.7
boiler.cc /usr/local 1 288 288 -- 0:09 31.2 0:04 86.3
boiler.cc /var 1 4256 4256 -- 0:21 205.9 0:04 1104.3
egbert.cc / 1 41952 41952 -- 1:26 487.3 0:37 1149.4
egbert.cc /opt 1 224 224 -- 0:06 37.5 0:02 136.0
egbert.cc -laris/install 1 64 64 -- 0:11 5.8 0:02 49.5
gurgi.cc. / 0 FAILED -----
gurgi.cc. /var 0 FAILED -----
pete.cc.p / 1 13408 13408 -- 0:41 328.2 0:08 1600.5
pete.cc.p /opt 1 3936 3936 -- 1:04 61.2 0:03 1382.6
pete.cc.p /usr 1 1952 1952 -- 0:29 67.0 0:03 584.3
pete.cc.p /var 1 300768 300768 -- 2:33 1963.8 2:50 1768.8
pete.cc.p /var/mail 0 6201184 6201184 -- 73:45 1401.3 73:47 1400.8
...
(brought to you by Amanda version 2.4.1p1)
```

Esta sección (que ha sido abreviada) reporta cada área copiada, mostrando el cliente, el área, nivel de copia, tamaños, tiempo de copia y tiempo de escritura a cinta. Las entradas están en orden alfabético por cliente y luego por área. Esto no es lo mismo que el orden de cinta. El orden de cinta puede ser determinado por la subopción `find` o `info` del comando `amadmin`, `amtoc` puede generar una tabla de contenidos de cinta tras cada ejecución, o `amreport` puede generar una lista impresa. Por defecto, los nombres de los clientes con truncados por la derecha, los nombres de área por la izquierda, para mantener el reporte con menos de 80 caracteres.

Dos archivos de registro son creados durante una ejecución de Amanda. Uno es llamado `amdump.NN`, donde `NN` es un número secuencial (1 es el más reciente, 2 es el siguiente más reciente, etc), y se encuentra en el mismo directorio que `amanda.conf`. El archivo contiene información detallada sobre la ejecución y es usado para estadísticas por `amplot` y `amstatus`, y también para depuración de errores. El otro archivo es llamado `log.YYYYMMDD.N`, donde `YYYYMMDD` es la fecha de la ejecución de Amanda, y `N` es un número secuencial, en el caso de que más de una ejecución se haya realizado el mismo día (0 para la primera ejecución, 1 para la segunda, etc). Este archivo está en el directorio especificado por el parámetro `logdir` del archivo `amanda.conf`. Contiene un sumario de la ejecución y es la base para el reporte que recibimos por correo electrónico. De hecho, `amreport` puede ser ejecutado manualmente para regerar un reporte en base a un archivo de registro antiguo.

Lo viejos archivos `amdump.NN` son eliminados por el script `amdump`. Los viejos archivos `log.YYYYMMDD.N` no son automáticamente eliminados, y deberían ser eliminados periódicamente a mano. Mantener un ciclo completo de cinta es una buena idea. Si el ciclo de la cinta es 40 y Amanda se ejecuta una vez al día, el siguiente comando haría el trabajo:

```
# find log.???????.* -mtime +40 -print | xargs rm
```

Si la opción `--with-pid-debug-files` fue usada en el `./configure`, los clientes acumularán archivos de depuración de errores en `/tmp/amanda` (o donde lo hayamos configurado con `--with-debug`) y deberían ser eliminados periódicamente. Sin esta opción, los archivos de depuración de los clientes tienen nombres fijos, y son rehusados de una ejecución a otra.

http://www.sergio-gonzalez.com/personales/ingenieria_informatica/sistemas_informaticos/documentacion/amanda/amanda.html

Rsync

Rsync⁶ es una aplicación para sistemas de tipo Unix que ofrece transmisión eficiente de datos incrementales comprimidos y cifrados. Mediante una técnica de delta encoding, permite sincronizar archivos y directorios entre dos máquinas de una red o entre dos ubicaciones en una misma máquina, minimizando el volumen de datos transferidos.

Algoritmo

El algoritmo en que se basa rsync fue creado por el programador australiano Andrew Tridgell, permite transmitir eficientemente una estructura a través de un canal de comunicación cuando el receptor ya tiene una versión diferente de la misma estructura. Su funcionamiento a grandes rasgos es el siguiente:

- El receptor divide su copia del archivo en bloques de tamaño fijo S , no solapados, y calcula dos sumas de verificación (checksum) para cada bloque: el hash MD4, y un 'rolling checksum' más débil. Luego envía estos checksums al emisor.
- El emisor calcula el rolling checksum para cada posible bloque de tamaño S , solapando bloques. Esto se puede hacer eficientemente gracias a una propiedad del rolling checksum: si el rolling checksum de los bytes n a $n+S-1$ es R , se puede calcular el checksum de los bytes $n+1$ a $n+S$ a partir del byte n , el byte $n+S$ y R , sin tener que examinar los bytes intermedios
- El emisor compara entonces sus rolling checksums con los enviados por el receptor para determinar si existen coincidencias. Cuando las encuentra, verifica calculando el checksum MD4 para el bloque propio coincidente y comparando con el del bloque del receptor.
- Finalizado el proceso el emisor envía únicamente los bloques para los cuales no se encontró ninguna coincidencia, junto con instrucciones para su ensamblado en la versión del receptor.

Características y aplicaciones

En adición a este algoritmo, la aplicación rsync provee otras funciones que asisten en la transferencia. Estas incluyen compresión y descompresión de los datos bloque por bloque, utilizando zlib, y soporte para protocolos de cifrado, tal como SSH.

Adicionalmente puede utilizarse una aplicación de tunneling para asegurar los datos.

Además de archivos, el algoritmo permite copiar directorios, aún recursivamente, así como vínculos, dispositivos, grupos y permisos. No requiere por defecto privilegios de root para su uso.

Los principales usos de rsync incluyen espejado (mirroring) o Respaldo de múltiples clientes Unix dentro de un servidor Unix central. Habitualmente se lo ejecuta mediante herramientas de scheduling como cron, para automatizar procesos de sincronización

Uso de la aplicación

La invocación más simple de la aplicación a través de línea de comandos tiene la siguiente forma:

```
rsync [OPTION]... SRC [SRC]... DEST
```

⁶ <http://es.wikipedia.org/wiki/Rsync>

La simplicidad de la aplicación se puede ver en el siguiente ejemplo (se utilizan opciones largas para facilitar su explicación, pero también se pueden utilizar opciones abreviadas):

```
rsync --verbose --compress --rsh=/usr/local/bin/ssh --recursive \  
--times --perms --links --delete --exclude "*bak" --exclude "*~" \  
/www/* webserver:/www
```

Este comando ejecuta rsync en modo verboso (muestra mensajes de estado por la salida estándar), con compresión, a través de ssh, en forma recursiva para los subdirectorios, preservando fechas y permisos del archivo origen, incluyendo vínculos, borrando archivos que fueron borrados en el directorio origen, excluyendo backups y archivos temporales (*bak, *~); el origen es el contenido del directorio /www y el destino el directorio /www en el host webserver.

Adicionalmente es necesario configurar uno de los puntos hosts como servidor rsync, ejecutando rsync en modo daemon:

```
rsync --daemon
```

y configurando el archivo `/etc/rsyncd.conf`.

Una vez configurado un servidor, cualquier máquina con rsync instalado puede sincronizar archivos hacia o desde éste.

Variantes

Existen otras utilidades basadas en el algoritmo rsync. `rdiff` genera archivos delta con la diferencia entre dos archivos, que puede ser aplicada a uno para convertirlo en el otro en cualquier momento. **rdiff-backup** utiliza `rdiff` para mantener espejos de backup de un directorio a través de una red. `Rdiff-backup` almacena delta `rdiff` incrementales, lo que permite recrear el directorio en cualquier punto de backup. Existen además versiones para otros sistemas operativos, tal como **rsyncX**, una versión para Mac OS X que permite transferir forks de recursos, y que en su versión **rsyncXCD** permite además crear particiones booteables. Incluso es posible utilizar rsync en Windows, a través de Cygwin.

FlyBack 0.4.0

Copias de seguridad al estilo TimeMachine de Apple ⁷

Una de las novedades de Leopard, la última versión de Mac OS X, es TimeMachine, que "fotografía" tu disco duro y permite ver y recuperar el contenido de hace unos días o unas semanas.

FlyBack recoge esta filosofía, y con un entorno más sencillo y sin artificios gráficos guarda copias de seguridad de tus ficheros, manualmente o de manera automática.

Con FlyBack podrás ver qué había en una carpeta concreta determinado día y así saber los cambios que has realizado en el tiempo y recuperar archivos borrados por accidente.

Para utilizar FlyBack necesitas:

- Sistema operativo: X11

Requisitos mínimos:

- python
- python-glade2
- python-gnome2
- python-sqlite3
- python-gconf
- rsync

FlyBack utilidad de backup para Linux

Flyback es una utilidad similar al TimeMachine para Mac, así como lo es también TimeVault.

Flyback⁸ esta basado en la herramienta de backup rsync. Crea espejos de directorios backups sucesivos del file que quieres respaldar, pero no cambia el link del backup previo. Esto previene el desperdicio de espacio en el disco mientras provee un acceso completo a todos los archivos sin ningun otro programa de recuperación (recovery). Si tu maquina se malogra, solo mueve tu disco externo a tu nueva maquina y copia el ultimo backup usando tu explorador de archivos favorito.

Nota que esto significa que puedes borrar selectivamente específicos backups y seguir reteniendo los archivos guardados (puedes borrar el backup del jueves y mantener el del lunes, sin desechar el del Miércoles).

Instalar Flyback es muy fácil, primero resolvemos las dependencias:

Debian	\$ sudo apt-get install python python-glade2 python-gnome2 python-sqlite rsync
Ubuntu	\$ sudo apt-get install python python-glade2 python-gnome2 python-sqlite python-

⁷ <http://flyback.softonic.com/linux>

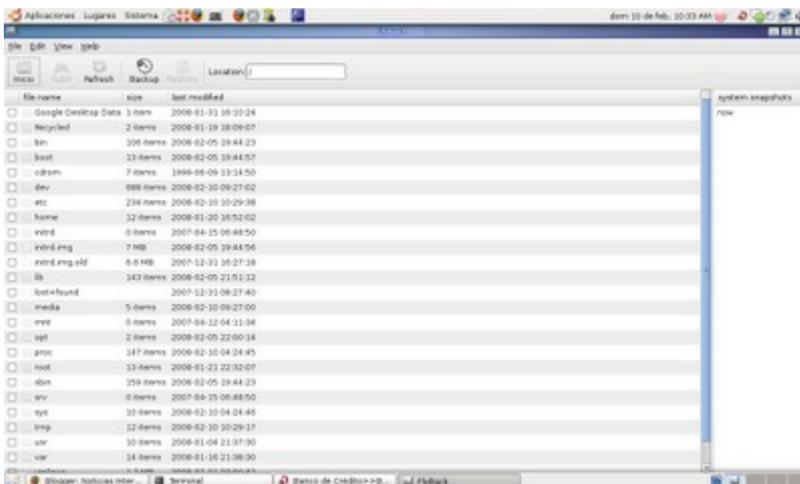
⁸ <http://comulinux.blogspot.com/2008/02/flyback-utilidad-de-backup-para-linux.html>

```
gconf rsync
```

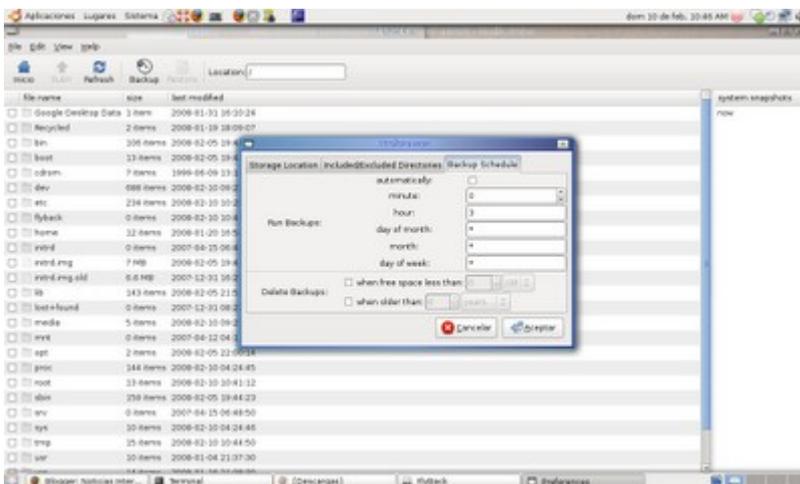
Redhat/Fedora

```
$ yum install pygtk2 gnome-python2-gconf pygtk2-libglade python-sqlite
```

Luego descargar el archivo Flyback <http://code.google.com/p/flyback> y desempaquetarlo usando:
`tar -zxvf flyback_0.4.0.tar.gz`
 Ingresar al directorio **`flyback`** y ejecutar desde un terminal ejecutar:
`python flyback.py`



Para configurarlo, escoger la carpeta al que se hará el backup, cuantas veces, etc.



El * significa todos los días, meses y años. */2 significa que hará el backup cada 2 horas. Para hacer el backup se recomienda un disco duro externo.

Backerupper 0.24

Realiza copias de seguridad periódicas de tus carpetas importantes ⁹

Backerupper es una sencilla herramienta que copia el contenido de la carpeta que escojas a otra carpeta cada cierto tiempo.

Es muy fácil de configurar. Creas un perfil, escoges una carpeta a copiar, el destino y la frecuencia de copia. Backerupper hará el resto.

Backerupper es muy útil para tener una copia de seguridad de tu carpeta de usuario o de la carpeta en la que guardes tus documentos imprescindibles.

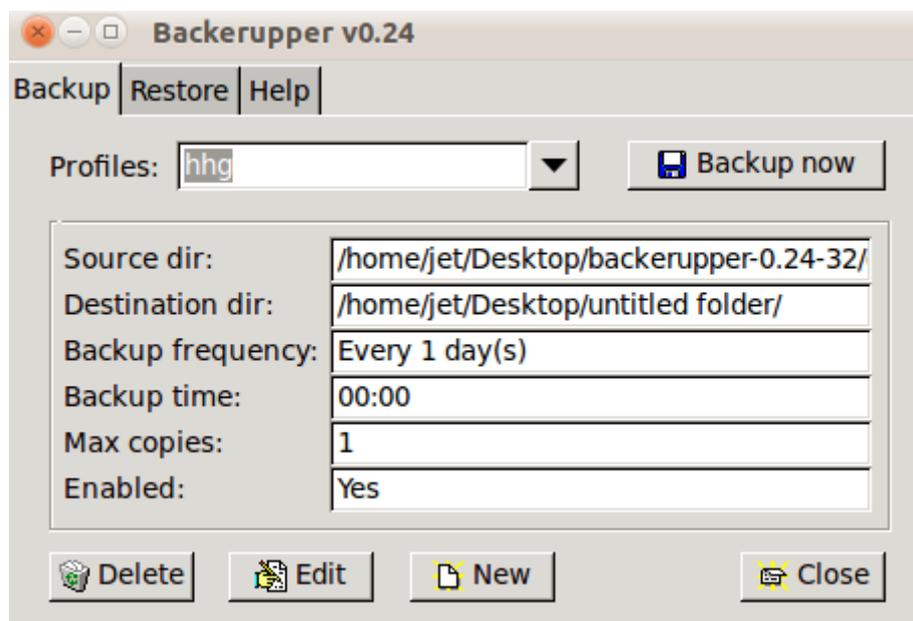
Backerupper es una sencilla herramienta que copia el contenido de la carpeta que escojas a otra carpeta cada cierto tiempo.

Es muy facil de configurar. Creas un perfil, escoges una carpeta a copiar, el destino y la frecuencia de copia. Backerupper hara el resto.

Backerupper es muy util para tener una copia de seguridad de tu carpeta de usuario o de la carpeta en la que guardes tus documentos imprescindibles.

Para utilizar **Backerupper** necesitas

- Sistema operativo X11 Gnome
-



⁹ <http://backerupper.softonic.com/linux>



Backerupper¹⁰ es una herramienta libre para realizar copias de seguridad instantáneas en Linux. Si se permite la analogía, una especie de **TimeMachine linuxero**.

Podemos configurar Backerupper para que, por ejemplo, realice de forma automática copias de seguridad de los datos que deseemos cada 3 horas, pero podemos ajustar el periodo de actualización de las copias en función de nuestras necesidades.

Y si necesitamos restaurar los datos, pues sólo tenemos que seleccionar la copia de seguridad que nos interesa y en un solo clic todo vuelve a estar en su sitio. En **Howto forge** hay una guía para su instalación y uso en Ubuntu.

Las copias de seguridad son algo imprescindible. Hay que tomárselas en serio, sobre todo cuando tenemos datos importantes en nuestro disco duro. Por último un consejo: **siempre que sea posible hay que automatizar el proceso de copias de seguridad**. La experiencia nos ha enseñado que, si no se automatizan, no se hacen.

¹⁰ <http://bitelia.com/2008/03/14/descarga-del-dia-backerupper/>

Fuentes de información:

<http://bacula.softonic.com/linux>

<http://es.wikipedia.org/wiki/Bacula>

<http://libertonia.escomposlinux.org/story/2004/5/5/93545/27085>

http://es.wikipedia.org/wiki/Advanced_Maryland_Automatic_Network_Disk_Archiver#Principales_caracter.C3.ADsticas

http://www.sergio-gonzalez.com/personales/ingenieria_informatica/sistemas_informaticos/documentacion/amanda/amanda.html

<http://lopezivan.blogspot.com/2007/05/rsync-vs-backuppc-vs-link-backup.html>

<http://es.wikipedia.org/wiki/Rsync>

<http://flyback.softonic.com/linux>

<http://comulinux.blogspot.com/2008/02/flyback-utilidad-de-backup-para-linux.html>

<http://backerupper.softonic.com/linux>

<http://administracion-linux.nireblog.com/post/2008/06/27/backerupper-024>

<http://bitelia.com/2008/03/14/descarga-del-dia-backerupper/>

