

Visualizar el rendimiento de sistemas con MRTG

# DE UN VISTAZO

MRTG genera gráficos sencillos para visualizar el rendimiento de una red de un vistazo.

**POR MATTHEW D. SACKS**

Jeff Metzger, Fotolia

Los administradores de Linux a menudo se benefician de poder capturar métricas del rendimiento del sistema, como pueden ser el uso de disco, de la CPU y de la memoria. Un gráfico sencillo nos puede ayudar a diagnosticar problemas y a analizar contratiempos en el tráfico de la red.

Multi-Router Traffic Grapher (MRTG) [1] nos permite recabar y visualizar en un gráfico la información de la red y del rendimiento de un servidor Linux. Es una herramienta para gráficos desarrollada en software libre que recupera y muestra estadísticas a partir de dispositivos de red basados en SNMP.

Las herramientas comerciales, como OpenView de HP, o Tivoli de IBM, así como algunas utilidades de software libre, como Cacti y Zenoss (una versión aumentada de MRTG), juegan un papel similar, pero en mi opinión, MRTG es la mejor solución para recabar y ver las tendencias a partir de la información de rendimiento del servidor.

## Configuración

Configurar MRTG y SNMP no es para cobardes. Especialmente si optamos por compilar todos los paquetes de software necesarios desde sus fuentes. Podemos verificar nuestro sistema de manteni-

miento de paquetes para ver si los paquetes MRTG están disponibles en nuestra distribución. En caso contrario, encontraremos el código fuente en la página Web del creador de MRTG, Tobi Oetiker [1].

## Simple Network Monitoring Protocol

SNMP es el protocolo que hay detrás de los mágicos gráficos que dibuja MRTG. En este artículo vamos a suponer que tenemos un conocimiento básico de SNMP. En caso contrario, en Internet hallaremos

abundante información del protocolo SNMP, quizá algo arcaico pero increíblemente potente. Para más información relativa a cómo poner en marcha SNMP en una máquina Linux, puede acudir a [2].

## Instalar MRTG

En el cuadro titulado “Requisitos Previos de MRTG” encontraremos información de los componentes que necesita MRTG. Una vez que los hayamos instalado, descargamos el código fuente de MRTG [1].

### Requisitos Previos de MRTG

Antes de intentar instalar MRTG debemos asegurarnos de tener lo siguiente:

- GCC – El compilador GNU C [3] viene preinstalado en la mayoría de las distros Unix y Linux. Si no lo tenemos disponible por defecto, debemos buscar el paquete usando el sistema de administración de paquetes.
- Perl – Gran parte del sistema MRTG está escrito en el lenguaje de script Perl. Debemos asegurarnos de que tenemos una versión reciente instalada (pruebe con `perl -v`). Se necesita al menos la versión 5.005 para que MRTG funcione correctamente. Si usamos SNMPV3 y otras funcionalidades nuevas, debemos usar al menos la 5.8.
- GD – La librería de generación de gráficos GD creada por Thomas Boutell [4]. Nótese que todas las versiones posteriores a la 1.3 sólo crean imágenes PNG. Thomas se metió en problemas debido a que formato GIF que solía usar GD requería una tecnología de compresión patentada por Unisys. MRTG puede funcionar tanto con las versiones antiguas como con las nuevas de la librería GD.
- libpng – GD las requiere para poder generar los archivos gráficos PNG [5].
- zlib – libpng necesita zlib para poder comprimir los archivos gráficos que hayamos creado [6].

Descomprimos y desempaquetamos el paquete de instalación desde un directorio temporal de instalación, y ejecutamos los siguientes comandos:

```
tar -xvzf mrtg.tar.gz
cd mrtg-2.15.2
./configure --prefix=/usr/local/mrtg-2.15.2
```

Puede que queramos hacer un enlace simbólico para referenciar la última compilación, lo que nos permitirá actualizar fácilmente o probar nuevas versiones con sólo cambiar el enlace simbólico:

```
ln -s /usr/local/mrtg-2.15.2 /usr/local/mrtg
```

Si el script de configuración no puede encontrar alguna dependencia, es probable que obtengamos un error parecido al que se muestra en el Listado 1.

Puede ser una buena idea descargar todas las últimas versiones de las librerías de las que el script de configuración de MRTG se esté quejando, incluso si aparentemente ya estuvieran en el equipo. MRTG tendrá entonces acceso al último código con (esperemos) menos bugs.

## Configurar la Instancia MRTG

Para compilar MRTG nos ubicamos en el directorio con las fuentes de instalación de MRTG:

```
cd /tmp/mrtg_source
./configure --prefix=/usr/local/mrtg-2.15
```

Si alguna dependencia de librerías se compiló desde la fuente, debemos modificar nuestros parámetros de configuración del script para incluirlas. Por ejemplo, si la librería GD software se descargó e instaló desde la fuente, modificamos nuestro script de configuración de la siguiente manera:

```
./configure --prefix=/usr/local/mrtg-2.15
--with-gd-lib=/usr/local/gd-2.0.34
--with-gd-inc=/usr/local/gd-2.0.34/lib
make
make install
```

para incluir la librería GD.

### Tabla 1: Opciones Línea Comandos de cfmaker

Opción	Descripción
community	Define el nombre comunitario SNMP.
global	Define los parámetros de configuración globales para cada servidor configurado.
Workdir	Directorio donde se va a guardar el HTML y las imágenes con los gráficos (preferiblemente en el directorio de contenido de nuestro servidor Web).
output	Define dónde se va a crear el archivo de configuración de MRTG. Añadimos los nombres de los servidores a consultar, o sus direcciones IP, al final del script cfmaker, separados por espacios. En nuestro ejemplo, sólo tenemos un servidor al que consultar periódicamente, Tux.

## En Busca de Rendimiento

En este artículo usaremos dos equipos Linux como ejemplo: Tux y Grapher. Tux es el equipo Linux del que queremos recabar y visualizar información del rendimiento, y Grapher es el equipo Linux que ejecuta MRTG.

Los archivos de configuración de MRTG son complejos y engorrosos para editar a mano, razón por la cual MRTG viene con un script de configuración muy fácil de usar. El script de configuración *cfmaker* ofrece muchas opciones. Vamos a usarlo para rellenar el archivo *mrtg.conf* con la información necesaria para generar gráficos básicos. Los argumentos en línea de comandos más básicos se muestran en la Tabla 1. Antes de ejecutar el script *cfmaker* creamos un directorio para albergar los archivos de configuración de MRTG, ya que puede que estemos usando dife-

rentes configuraciones con un mismo equipo:

```
mkdir /usr/local/mrtg/cfg
```

Ejecutamos el script *cfmaker* de la siguiente manera:

```
/usr/local/mrtg/bin/cfmaker
--community=public --global
Options[_]:
growright,avgeak,printrouter
--global Workdir:
/usr/local/apache2/htdocs
--output=/usr/local/mrtg
/cfg/mrtg2.cfg Tux
```

Como he explicado anteriormente, MRTG recibe la información mediante SNMP. El protocolo SNMP organiza el direccionamiento de la información del dispositivo en una estructura jerárquica conocida

### Listado 1: Problemas de Configuración con MRTG

```
01 ** Oops, one of many bad things happened:
02 a) You don't have the GD library installed.
03 Get it from http://www.boutell.com, compile it, and
04 use either --with-gd-lib=DIR or --with-gd-inc=DIR to specify
05 its location. You might also have to use --with-z-inc,
06 --with-z-lib and --with-png-inc, --with-png-lib for gd
07 versions 1.6 and higher. Check config.log for more
08 information on the problem.
09 b) You have the GD library installed, but not the gd.h
10 header file. Download the source (see above) and use
11 --with-gd-inc=DIR to specify where the file can be found.
12 c) You have the library and the header file installed, but
13 you also have a shared GD library in the same directory.
14 Remove the shared library files and/or links (e.g.,
15 libgd.so.2.0.0, libgd.so and libgd.so.2). This is especially
16 likely if you're using a recent (post 1.8.4) version of GD
17 and didn't configure it with --disable-shared.
18 d) You have the GD library installed and also its headers, but
19 you are
20 missing libpng (and headers) or freetype (and headers)
21 (MRTG does not use freetype, but if your copy of GD is
22 precompiled
23 against it, you have to install it ... )
```

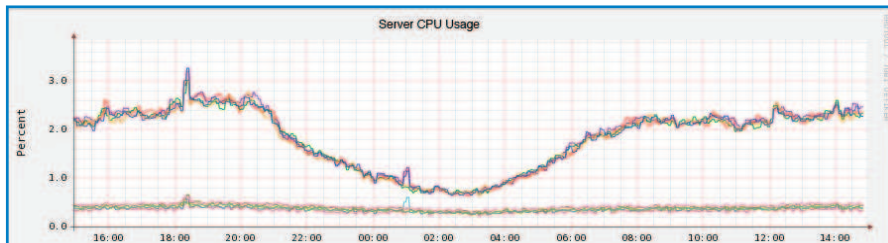


Figura 1: Un gráfico de uso de CPU con un patrón normal.

como Management Information Base (MIB). El siguiente ejemplo supone que la UCD-SNMP MIB está cargada y que la información puede ser consultada periódicamente haciendo uso de esta definición MIB. Para comprobar si está disponible o no la UCD-SNMP MIB, verificamos nuestra instalación SNMP para ver si está cargada.

## Test UCD-SNMP MIB

Una manera rápida de comprobar si tenemos disponible la UCD-SNMP MIB es usar el siguiente comando contra el servidor Linux al que solicitamos información:

```
snmpwalk -v1 -c public hostname ssCpuRawUser
```

Este comando consulta la cifra de uso de CPU por usuario en el equipo que es nuestro objetivo.

Si falla este comando, verificamos si SNMP se está ejecutando en el equipo objetivo, debiéndonos asegurar de que la sintaxis del comando `snmpwalk` sea correcta (véase la página man de

`snmpwalk`). Igualmente, comprobamos que la UCD-SNMP MIB esté instalada.

## Monitorizar el Uso de CPU

Ahora que ya hemos generado un archivo `mrtg2.cfg` estándar mediante el script `cfgmaker`, debemos editar a mano `mrtg2.cfg` para poder renderizar las estadísticas de recursos como uso de la memoria, del disco y de la CPU.

La página Web de MRTG tiene una excelente documentación acerca de todos los parámetros para configurar MRTG.

Este ejemplo describe cómo crear una sencilla plantilla para dibujar el uso de la CPU.

La UCD-SNMP MIB nos permite monitorizar un amplio rango de configuraciones relacionadas con el rendimiento. Para ver las que tenemos disponibles, podemos acudir a la definición del MIB [7].

Para añadir un parámetro como el uso de la CPU a la configuración de MRTG, comenzamos por crear los siguientes directorios que alberguen estas nuevas configuraciones a medida:

```
mkdir /etc/mrtg
mkdir /etc/cron.mrtg
```

A continuación creamos `cpu.cfg` para monitorizar la carga de la CPU haciendo uso del contenido mostrado en el Listado 2. Creamos un archivo cron para el trabajo de monitorizar la CPU.

```
vi /etc/cron.mrtg/cpu
```

con el siguiente contenido:

```
#!/bin/sh
env LANG=C /usr/local/mrtg/bin/mrtg /etc/mrtg/cpu.cfg
```

Con el comando que sigue hacemos que el archivo `cpu.cfg` sea ejecutable:

```
chmod +x /etc/cron.mrtg/cpu
```

Para generar información gráfica, ejecutamos el script unas tres veces:

```
sh /etc/cron.mrtg/cpu
```

(Puede que nos aparezcan algunos avisos, que podemos ignorar sin problemas). Para ver los resultados, creamos un archivo de índice con el script `indexmaker`:

```
indexmaker -output=/usr/local/apache2/htdocs/mrtg/cpu_index.html -title="CPU Usage" -sort=name -enumerate /etc/mrtg/cpu.cfg
```

Por último, añadimos un super cron que contenga los OIDs específicos:

```
*/* * * * * /bin/run-parts /etc/cron.mrtg 1> /dev/null
```

Para dibujar otras métricas del sistema, como el uso de memoria, usuarios medios y uso del disco, podemos consultar la definición del UCD-SNMP MIB [7] y sim-

## Listado 2: cpu.cfg

```
01 ## Graph Tux CPU ##
02 WorkDir: /usr/local/apache2/htdocs/mrtg
03 LoadMIBs: /usr/share/snmp/mibs/UCD-SNMP-MIB.txt
04 Target[Tux.cpu]:ssCpuRawUser.0&ssCpuRawUser.0:linux-magazine@Tux +
05 ssCpuRawSystem.0&ssCpuRawSystem.0:public@Tux +
06 ssCpuRawNice.0&ssCpuRawNice.0:public@Tux
07 RouterUptime[Tux.cpu]: public@Tux
08 MaxBytes[Tux.cpu]: 100
09 Title[Tux.cpu]: CPU Load
10 PageTop[Tux.cpu]: <H1>Active CPU Load %</H1>
11 Unscaled[Tux.cpu]: ymwd
12 ShortLegend[Tux.cpu]: %
13 YLegend[Tux.cpu]: CPU Utilization
14 Legend1[Tux.cpu]: Active CPU in % (Load)
15 Legend2[Tux.cpu]:
16 Legend3[Tux.cpu]:
17 Legend4[Tux.cpu]:
18 LegendI[Tux.cpu]: Active
19 LegendO[Tux.cpu]:
20 Options[Tux.cpu]: growright,nopercent
```

## Agradecimientos

Damos las gracias a Tobi Oetiker por dar permiso para reproducir la información de la instalación y los requisitos previos, que hemos obtenido de su página Web [1], y por crear esta herramienta de redes tan útil. Agradecemos también a Safdar Husain por sacar tiempo para mostrarnos el mundo de la Administración de Sistemas.

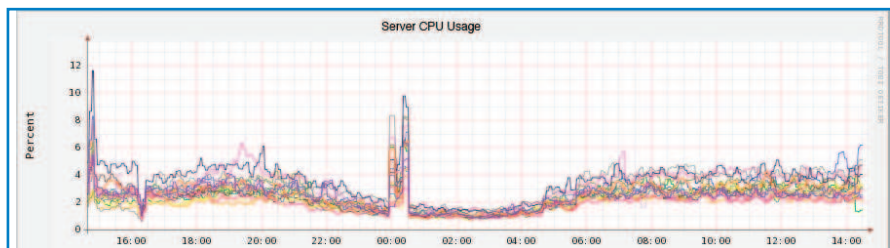


Figura 2: Un pico anormal aparece en este gráfico de uso de CPU.

plemente repetir los pasos precedentes, modificando el OID y dibujando los parámetros comentados.

## Consulta Periódica

Ya hemos creado una plantilla para configurar gráficos a medida de los recursos del sistema con objeto de mejorar el rendimiento. El siguiente paso es elegir un intervalo de consulta para recabar la información. Lo haremos proporcionando la ruta al binario de MRTG y ejecutándolo en los intervalos deseados mediante cron.

Para editar el crontab, tecleamos *crontab -e* y añadimos la siguiente entrada:

```
0,5,10,15,20,25,30,35,40,
45,50,55 * * * * env LANG=C
/usr/local/mrtg/bin/mrtg
/usr/local/mrtg/cfg/mrtg2.cfg
```

Esta entrada hará que se ejecute MRTG cada cinco minutos y poblará los gráficos. Si el cron job no está configurado para ejecutar el binario MRTG, el gráfico no se rellenará. Podemos ajustar el intervalo de consulta a nuestro gusto, pero no debemos olvidar que con intervalos poco frecuentes el gráfico será menos preciso.

Dependiendo de cómo esté configurado su servidor Web, puede que tengamos que referenciar el nombre de archivo específico que haya sido generado por los scripts MRTG.

En este ejemplo, se usó Apache y los archivos se generaron en el directorio WorkDir.

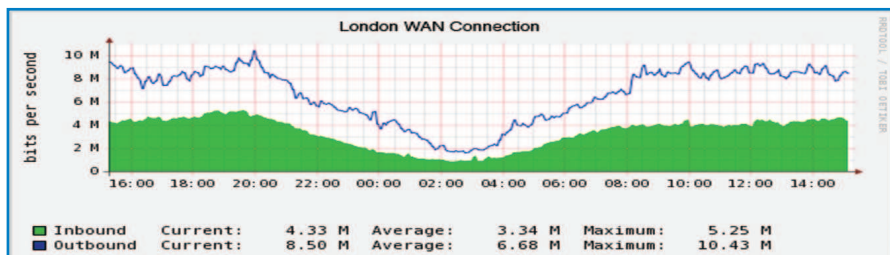


Figura 3: Un rápido vistazo al gráfico de uso de la red muestra que el pico no ha sido causado por tráfico de red.

## Mostrar Gráficos de Rendimiento

La herramienta *indexmaker* creó un archivo de índice, que encontraremos en `/usr/local/apache2/htdocs/mrtg/cpu_index.html`. Sólo tenemos que abrir esta URL en un navegador para ver el gráfico de la CPU.

## Mostrar Gráficos de Ancho de Banda

Los nombres de archivo a veces se generan automáticamente, por lo que es mejor mirar en el directorio de salida en busca del nombre de archivo, y luego introducirlo como URL en el navegador.

Después de todo este duro trabajo, deberíamos tener unos elaborados gráficos y un resumen del comportamiento del sistema con un sólo vistazo. Si volvemos a ejecutar *indexmaker* o diseñamos un portal que muestre gráficos PNG generados dinámicamente, podemos escalar nuestra instalación MRTG para incluir métricas de rendimiento e información de la infraestructura de toda una empresa.

RRDtool, la herramienta de base de datos round robin [8], nos permite tener un mejor rendimiento y mayor nivel de personalización con los gráficos de MRTG. Debemos usar RRDtool cuando instalemos MRTG para un gran número de servidores Linux.

## Interpretar los Gráficos

En general, los gráficos MRTG son más efectivos cuando se observan e interpretan regularmente. Es mejor guardar algún

tipo de gráfico base para usarlo como comparación cuando haya que solucionar algún contratiempo o estemos investigando problemas.

Por ejemplo, la Figura 1 muestra un grupo de servidores con una carga normal. (Nótese que la capacidad de añadir múltiples servidores en un único gráfico como éste es una funcionalidad de RRDtool, que es un añadido de MRTG). La mayor parte del tiempo se observa en el gráfico un patrón consistente a lo largo del tiempo.

Si aparece un pico a medianoche en el gráfico de la CPU del servidor (véase la Figura 2), una posible explicación puede ser algún trabajo de copia de seguridad planificado para esta hora. Otra posibilidad es que esté ocurriendo algún tipo de ataque a la granja de servidores. Para poder discernir entre estos dos escenarios, podemos comparar las gráficas de uso de la CPU con las de tráfico de red (véase la Figura 3).

El tráfico de red no muestra ningún pico a la misma hora en que lo muestra el uso de la CPU, por lo que podemos asegurar que no se trata de un ataque externo a nuestros servidores. El siguiente paso sería investigar el servidor o los logs de copia de seguridad para ver si había algún trabajo programado para esa hora.

## Conclusión

MRTG permite al administrador investigar rápidamente posibles cambios en el rendimiento del sistema.

En este artículo hemos mostrado sólo una pequeña fracción de las posibilidades para presentar en gráficos información del rendimiento del sistema (y arreglar problemas) con MRTG. ■

## RECURSOS

- [1] MRTG: <http://oss.oetiker.ch/mrtg/>
- [2] Net-SNMP: <http://www.net-snmp.org>
- [3] GCC: <http://gcc.gnu.org>
- [4] GD library: <http://www.boutell.com/gd/>
- [5] libpng: <http://www.libpng.org/pub/png/libpng.html>
- [6] zlib: <http://www.gzip.org/zlib>
- [7] Definición de UCD-SNMP MIB: <http://www.oidview.com/mibs/2021/UCD-SNMP-MIB.html>
- [8] RRDtool: <http://oss.oetiker.ch/rrdtool>