



Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

Creación de gráficas con RRDTOOL (49152 lectures)

Per **Jesús Roncero Franco**, [golan](http://www.roncero.org) (<http://www.roncero.org>)

Creado el 19/04/2002 15:40 modificado el 19/04/2002 15:40

(1) RRDTOOL, que viene de Round Robin Databases, bases de datos circulares, es un sistema que permite almacenar y representar datos en intervalos temporales (ancho de banda, temperatura, ...). Guarda los datos en una base de datos que no crece en el tiempo y permite crear bonitas gráficas para representar los datos. Con este artículo veremos una introducción a la creación de bases de datos round robin y sus gráficas asociadas.

RRDTOOL es una re-implementación de [MRTG](#)⁽²⁾, un programa del mismo autor que nos permite tener gráficas del tráfico de datos a través de un dispositivo de red, una tarjeta de red, un router, etc... usando para ello el protocolo SNMP. RRDTOOL, lo que hace es aprovechar el mismo motor gráfico para implementar bases de datos Round Robin o bases de datos circulares.

Introducción

Una base de datos circular va a contener siempre la misma cantidad de datos, ya que funciona de manera que, cuando lleva almacenados toda la extensión de la base de datos, simplemente sobrescribe los datos antiguos. Pensad en ella como en un círculo en el que se van a ir colocando datos. Si empezamos en un punto colocando datos, cuando hayamos dado una vuelta a todo el círculo, habremos llegado al inicio del círculo, y es ahí cuando empezaremos a sobrescribir los datos que recopilamos al principio.

Una vez visto esto, nos estaremos preguntando para qué se puede usar este programa. Pues sirve para tener gráficas sobre cualquier cosa que pueda ser medida como por ejemplo la temperatura, el número de usuarios que están conectados por ssh a una máquina, la carga de una CPU o, como el programa MRTG hace, la tasa de transferencia de datos a través de un dispositivo de red.

Para que este artículo no se convierta en una traducción del tutorial y del manual crearemos una base de datos circular de ejemplo para medir la temperatura. En este caso, utilizo unos termómetros de los que leo los datos a través del puerto serie, los cuales los voy a introducir en la base de datos, pero vosotros podeis usar el dato que querais, por ejemplo la temperatura de vuestra CPU que podeis leer a través del paquete [lm-sensors](#)⁽³⁾.

Lo primero que teneis que hacer es instalar el programa. En debian es tan fácil como hacer un **'apt-get install rrdtool'**. En otros sistemas podeis buscar los paquetes RPM o ir directamente a la web de rrdtool y bajaros los fuentes para así compilarlos e instalarlos.

Una vez instalados ya podemos usar el programa rrdtool y consultar sus páginas del manual. Por ejemplo, si lo que queremos es crear una base de datos, la orden a ejecutar es **'rrdtool create'**, pero para consultar la página del manual, deberemos ejecutar **man rrdcreate'**.

Creación de una base de datos circular

Las bases de datos circulares se crean en un archivo con extensión **.rrd**. Cuando creamos un archivo **.rrd** podemos crear varias bases de datos dentro del archivo que se encargarán de sostener distintos datos. Por ejemplo, en nuestro ejemplo crearemos una base de datos que mantenga los datos de la temperatura de las últimas 24 horas, otra para la última semana, otra para el último mes y otra para el último año.



En el ejemplo que vamos a crear vamos a medir la temperatura proveniente de dos termómetros diferentes, en mi caso uno en el interior y otro en el exterior. Para ello, tenemos que definir dos **fuentes de datos (DS, data source)** y vamos a llevar la media, la máxima y la mínima tanto para los datos de las 24 últimas horas, de la última semana, el último mes y el último año. Veamos el comando para crearla:

```
rrdtool create temperatura.rrd \
-s 300 \
DS:escritorio:GAUGE:600:-10:60 \
DS:exterior:GAUGE:600:-10:60 \
RRA:AVERAGE:0.5:1:600 \
RRA:AVERAGE:0.5:6:700 \
RRA:AVERAGE:0.5:24:775 \
RRA:AVERAGE:0.5:288:797 \
RRA:MIN:0.5:1:600 \
RRA:MIN:0.5:6:700 \
RRA:MIN:0.5:24:775 \
RRA:MIN:0.5:288:797 \
RRA:MAX:0.5:1:600 \
RRA:MAX:0.5:6:700 \
RRA:MAX:0.5:24:775 \
RRA:MAX:0.5:288:797
```

Veamos el significado de este comando. En primer lugar, definimos el tiempo que va a pasar entre cada toma de datos con la opción **-s 300**. O sea, vamos a ir añadiendo datos cada 5 minutos. Estos van a ser los **puntos de datos primarios (primary data points)**.

A continuación definimos las fuentes de datos (**DS**) para el termómetro del interior y el del exterior, indicándole que va a ser de tipo **GAUGE** (ahora veremos otros tipos), 600 indica el tiempo máximo que puede pasar hasta que consideremos los datos como desconocidos. En este caso hemos puesto 10 minutos. Los siguientes parámetros son valores mínimos y máximos que se pueden registrar, o sea, -10°C y 60°C. Quizás dónde vivas si tengas que modificar estos valores ;-)

Los tipos de bases de datos que podemos tener son : **GAUGE** (Apropiado para temperaturas), **COUNTER**(apropiado para un contador), **DERIVE** (Calcula la derivada de la recta que va desde el último valor hasta el valor actual. Útil para saber la tasa de gente entrando en una habitación, por ejemplo), **ABSOLUTE** (Para valores absolutos que se ponen a 0 después de cada lectura, por ejemplo el número de mensajes recibidos en los últimos 5 minutos)

A continuación tenemos dónde definimos los datos que vamos a ir registrando.

RRA:CF:xff:steps:rows

Esto se hace con un **RRA (Round Robin Archive)**. Con RRA y sus parámetros indicamos qué tipo de dato queremos guardar. Para ello usamos una **función de consolidación (CF)** que va a ser la encargada de guardar los puntos de datos consolidados.

- **CF** puede ser **AVERAGE**, **MIN** y **MAX** guardan la media, el mínimo y la máxima como podemos intuir. En este caso ya son puntos de datos consolidados.
- **xff** es el *xfiles factor*, que es un factor, tal como explica en su página man, que sirve para ver qué parte de los datos se puede construir a partir de datos no válidos. Si os soy sincero, no entiendo bien el uso de esta parte y uso 0.5 que es el valor que viene en todos los ejemplos y el tutorial.
- **steps** indica cuantos de los datos primarios se necesitan para construir un dato consolidado.
- Y **rows** indica cuantos datos queremos que se guarden en el archivo.

Así,sabiendo que guardamos los datos cada 300s, para el primer ejemplo: **RRA:AVERAGE:0.5:1:600** le estamos diciendo que guarde la temperatura durante 50 horas (300s * 600s * 1 dato = 50 horas). En el segundo RRA indica que vamos a guardar la temperatura de 30 minutos (6 step * 300 seg = 30 minutos) durante 700 de estos intervalos, o sea, durante 350 horas (30 minutos * 700 intervalos = 350 horas). El tercer RRA le está indicando que guarde la temperatura (MAX,MED,MIN) de 2 horas (24 intervalos * 300 seg = 2 horas) y que lo guarde durante un par de meses (2 horas * 775 intervalos = 64.5 días) y el último RRA le está indicando que guarde la temperatura de 24 horas durante 797 días.

De esta manera podemos tener datos suficientes para crear gráficas del año (o más), del mes, semana o día para cada uno de los RRA que hemos definido.



¿difícil?

Introducción de los datos

Bien, llegados a este punto, tenemos que empezar a preguntarnos cómo meter los datos. La manera que tiene rrdtool de guardar los datos es emparejándolos con el tiempo unix estándar, es decir, almacenando cada par de valores (tiempo unix, dato a introducir), es decir, cada dato que se añada irá emparejado con un tiempo unix, es decir, el número de segundos que han pasado desde el 1 de enero de 1970. Por ejemplo, si quiero introducir las temperaturas 19°C y 34.5°C lo haré con la orden

```
rrdtool update temperatura.rrd N:19:34.5
```

En este caso las temperaturas se introduzcan en las fuentes de datos (DS) en orden, es decir el 19 para el primer DS y el 34.5 para el segundo, según como lo hayamos creado. En este caso, como hemos obviado el tiempo unix, rrdtool introducirá el tiempo unix actual. Este es el método más sencillo para poder hacer la actualización y con un breve script en bash podemos ir añadiendo los datos :

```
#!/bin/bash
# $escritorio y $exterior tienen los valores a introducir.
# Actualizar las BD
/usr/bin/rrdtool update /home/golan/datos/temperatura.rrd N:$escritorio:$exterior
```

En mi caso, en este script en perl leo los datos del programa que lee la temperatura del termómetro a través del puerto serie y almaceno los valores en las variables \$escritorio y \$exterior. Podéis sustituirlo por el programa de vuestra elección que os dé los datos que queráis introducir en la base de datos.

Una vez hecho esto y verificado el programa, añadimos una entrada al crontab de la siguiente forma (imaginemos que el script se llama /home/golan/bin/logtemp)

```
* /5 * * * * /home/golan/bin/logtemp
```

en el que le indicamos que se ejecute cada 5 minutos.

Los datos se pueden añadir también incluyendo el tiempo unix. Os recomiendo echarle un vistazo a la página man de *rrdupdate* para ver su sintaxis y ejemplos.

Consulta de datos

Una vez que ya tenemos recolectados unos cuantos datos, puede ser que queramos consultar los datos ya almacenados. Podemos hacerlo con el comando *fetch* que nos sacará por pantalla, para un determinado RRA, los datos almacenados. Por ejemplo, si ejecutamos

```
rrdtool fetch temperatura.rrd AVERAGE
```

obtendremos algo similar a esto:

```
1019178900: 2.3061600000e+01 1.6380000000e+01
1019179200: 2.3001600000e+01 1.6740133333e+01
1019179500: 2.3000000000e+01 1.6623466667e+01
1019179800: 2.2941800000e+01 1.6561800000e+01
1019180100: nan nan
```

dónde se indica, junto al tiempo unix, los valores almacenados en base de datos (cuando aparece **nan** o **NaN**, quiere decir que ese dato no es válido (Not A Number)). Podemos usar las opciones **--start** y **--end** para delimitar nuestros datos.

Si queremos hacer un backup de la base de datos, podemos obtener una copia en ASCII en formato **XML** con la que podremos regenerar la base de datos o usar los valores para otros análisis:

```
rrdtool dump temperatura.rrd > salida.xml
```



Creación de Gráficas

La creación de las gráficas es la parte, quizás, más complicada de esta herramienta, a la vez que es la más potente. Veamos como podemos generar algunas imágenes y despues veremos las opciones que nos brinda el sistema. Empezaremos con la gráfica de las últimas 24 horas. Para ello utilizo este sencillo shell script :

```
#!/bin/sh

rrd=/home/golan/datos/temperatura.rrd
last=`rrdtool last $rrd`
file1=/home/golan/datos/dia.png
fecha=`date +%c %Z`
width=600
height=250

# 1dia
/usr/bin/rrdtool graph $file1 --imgformat PNG --start -1day --end $last \
--width $width --height $height DEF:ext_c=$rrd:exterior:AVERAGE \
DEF:int_c=$rrd:escritorio:AVERAGE COMMENT:"          Terminus\\n" \
COMMENT:"          " COMMENT:"          Min          Max          Med          Último\\n" \
COMMENT:"          " GPRINT:ext_c:MIN:"Exterior %5.2lf C" \
GPRINT:ext_c:MAX:" %5.2lf C" GPRINT:ext_c:AVERAGE:" %5.2lf C" \
GPRINT:ext_c:LAST:" %5.2lf C\\n" COMMENT:"          " \
GPRINT:int_c:MIN:"Interior %5.2lf C" \
GPRINT:int_c:MAX:" %5.2lf C" GPRINT:int_c:AVERAGE:" %5.2lf C" \
GPRINT:int_c:LAST:" %5.2lf C\\n" COMMENT:"\\s" LINE1:ext_c\#FF8000:'Exterior' \
LINE1:int_c\#8080F0:'Interior' COMMENT:"Actualizado: $fecha \\c"
```

Veamos las opciones : la orden para crear el gráfico es **rrdtool graph**. Con **man rrdgraph** obtendremos más ayuda con las opciones. En el script guardaremos las siguientes variables :

- **rrd**: Contiene el path de la base de datos.
- **last**: El tiempo unix del último registro añadido a la base de datos.
- **file1**: El fichero de salida.
- **fecha**: La fecha actual para la generación de la gráfica.
- **width** y **height**: La anchura y altura de la gráfica.

Al usarlas con el comando de generación de la gráfica, se las paso al principio, como podeis ver, para generar una imagen con formato PNG, que comience un día antes que la fecha actual y que termine con la fecha del último registro. La especificación del inicio de los datos se puede hacer en tiempo negativo o en forma negativa. De esta forma, especificando **-1day** empezaremos un día atrás, **-7day** una semana atrás,... Estos parámetros son claros al igual que los parámetros de la altura y anchura del gráfico. Con el parámetro **DEF** lo que hacemos es definir un nombre virtual para una fuente de datos (**DS**) que podremos usar con las funciones que nos brinda rrdtool. Su sintáxis es :

DEF:vname=rrd:ds-name:CF

Dónde *vname* es el nombre que le damos, *rrd* es el fichero rrd, *ds-name* es el nombre de la fuente de datos (**DS**) que le hayamos dado en la creación de la base de datos rrd y *CF* es la función de consolidación que vamos a usar. En este caso *AVERAGE*, lo cual nos mostrará una gráfica con la temperatura del exterior media. Si hubiesemos querido una grafica con la temperatura maxima, habríamos usado la función de consolidación *MAX*.

Como veis, definimos una gráfica por cada fuente de datos que hemos definido usando la función de consolidación de la media. Con estos simples parámetros ya podemos generar unas bonitas gráficas, pero rrdtool nos permite una mayor funcionalidad usando los nombres virtuales de las *fuentes de datos* que hemos definido. Por ejemplo, para añadir comentarios, lo podemos hacer con la opción **COMMENT:"comentario\n "** generando un salto de línea con el famoso '\n'.

Con la opción **GPRINT** podemos escribir datos debajo de la gráfica en cuestión pasándole parámetros al estilo del printf de C. Su sintáxis es : **GPRINT:vname:CF:format** dónde *vname* es el nombre virtual que le hayamos definido, por ejemplo *ext_c*, *CF* es la función de consolidación y *format* es el formato que vamos a usar. En nuestro ejemplo *%5.2lf* nos indica que queremos 5 enteros como máximo con 2 decimales. Se pueden usar otros formatos que podeis consultar en la página *man rrdgraph*.

Por último tenemos la definición de las lineas de las gráficas. Esto lo hacemos con la opción

**LINE{1|2|3}:vname[#rrggb[:legend]]**

indicando también el grosor de esta. O sea, LINE1 creará una línea más fina que LINE2, etc. *vname* como es lógico indicará el nombre virtual sobre el que queremos hacer la gráfica, pudiendosele añadir, como opción, el color de la línea y una leyenda.

Otra opción interesante que no usamos en nuestro ejemplo es la opción **CDEF** con la que podemos crear fuentes de datos virtuales, en base a expresiones matemáticas y a partir de fuentes de datos (DS) que ya tengamos definidas. Por ejemplo, supongamos que quisieramos definir una gráfica que representase la diferencia entre la máxima y la mínima. La única pega a esto es que se utiliza la *notación polaca inversa o notación postfija*, es decir, el operador va después de los operandos. Si has manejado una calculadora científica HP de gama alta, sabras manejarte bien con este tipo de notación. Así, por ejemplo, lo que para nosotros, sumar dos números es **2+4**, para la notación postfija sería algo así como **2 4 +**. No voy a entrar en detalle sobre este tema ya que podreis encontrar multitud de textos sobre el tema en internet, incluyendo un buen tutorial en la página de rrdtool.

Bueno, veamos esto con el ejemplo de la diferencia entre la máxima y la mínima. La sintaxis para **CDEF** es :

CDEF:vname=rpn-expression

donde *vname* es el nombre virtual que vamos a usar para estos datos y *rpn-expression* es la expresión en notación polaca inversa que vamos a usar y que debe contener un nombre virtual que ya hayamos definido previamente con **DEF**. Así, por ejemplo, la instrucción :

```
/usr/bin/rrdtool graph $filemaxmin --imgformat PNG --start -7day --end \
$last --width $width --height $height \
DEF:ext_max=$rrd:exterior:MAX \
DEF:ext_min=$rrd:exterior:MIN \
CDEF:diferencia=ext_max,ext_min,- \
COMMENT:"          maxmin      \n" \
AREA:diferencia\#C760C5:'Diferencia' \
COMMENT:"Actualizado: $fecha \c"
```

genera el gráfico que buscabamos. Vemos que la expresión utilizada en '*diferencia*' es *ext_max,ext_min,-* que lo que hace, como ya debemos suponer es la diferencia entre la máxima y la mínima. Podemos apreciar aquí, que hemos usado otro tipo de gráfico, **AREA** en vez de **LINE** que lo que hace es rellenar de color desde 0 hasta el valor del dato. Si os fijais bien, este no es el tipo de representación que queremos hacer (no hace la diferencia entre la máxima y la mínima de un día), pero os da una idea aproximada de lo que se puede llegar a hacer con esta herramienta y un poco de imaginación.

Conclusión

Para terminar, podéis ver en mi página web todo esto funcionando, y con datos reales: <http://www.roncero.org/temp/>⁽⁴⁾. Cuando creéis vuestras bases de datos circulares con rrdtool vereis que es una herramienta muy versátil y que puede utilizarse para representar multitud de datos y en diversas situaciones, ¡solo hace falta ser creativo! :)

Lo que me queda decir es que experimenteis creando vuestras propias bases de datos, que os leais el manual y el tutorial donde se tratan temas más complicados para que tengais una idea de lo que se puede llegar a crear con este software.

Enlaces Relacionados

- Gráficas de ejemplo: <http://www.roncero.org/temp/>⁽⁴⁾
- Páginas de RRDTOOL: <http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/>⁽¹⁾
- Tutorial RRDTOOL: <http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/tutorial/rrdtutorial.html>⁽⁵⁾
- Tutorial CDEF: <http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/tutorial/cdeftutorial.html>⁽⁶⁾
- Tutorial Notación polaca inversa: <http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/tutorial/rpntutorial.html>⁽⁷⁾
- Programa para leer la temperatura y origen de parte de los scripts <http://www.brianlane.com>⁽⁸⁾

Lista de enlaces de este artículo:

1. <http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/>
2. <http://people.ee.ethz.ch/~oetiker/webtools/mrtg/>



3. <http://www.netroedge.com/~lm78/>
4. <http://www.roncero.org/temp/>
5. <http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/tutorial/rrdtutorial.html>
6. <http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/tutorial/cdefutorial.html>
7. <http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/tutorial/rpntutorial.html>
8. <http://www.brianlane.com>

E-mail del autor: jesus_ARROBA_roncero.org

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=1284>