

RRDtool, herramienta de bases de datos en round robin.

Enrique Galdú

galdu@si.uji.es

Tabla de contenidos

Introducción.....	3
Resultados	3
¿Cómo funciona internamente?	6
Funciones de RRDtool	8
Crear una base de datos RRD	9
Actualizar una base de datos RRD	12
Crear gráficos	13
Un ejemplo práctico.	14
Frontends: Remstats	17
Conclusiones.....	18
Mas información	18

Introducción.

¿Qué es RRDtool?

- RRDtool es un sistema para almacenar y mostrar datos a través del tiempo. Ej. tráfico de red, temperatura de la sala de máquinas, carga de servidores.
- Los datos se almacenan de manera compacta, round robin
- La base de datos no crece con el tiempo.
- Se puede mostrar fácilmente en forma de gráficos para distintos periodos de tiempo.

¿Qué se puede hacer con esta herramienta?

- cualquier cosa que se te ocurra

Necesitas:

- poder medir un valor a lo largo del tiempo
- especificar que resoluciones deben guardarse
- alimentar a RRDtool con estos valores
- crear gráficos: PNG, GIF (con esta herramienta)
- publicarlos en páginas web

El Servei d'Informàtica lo utiliza sobre su infraestructura:

- monitorización
- estudio de rendimiento
- disponibilidad
- estadísticas de uso
- alertas

Resultados

Vamos a ver algunos gráficos

RRDtool, herramienta de bases de datos en round robin.

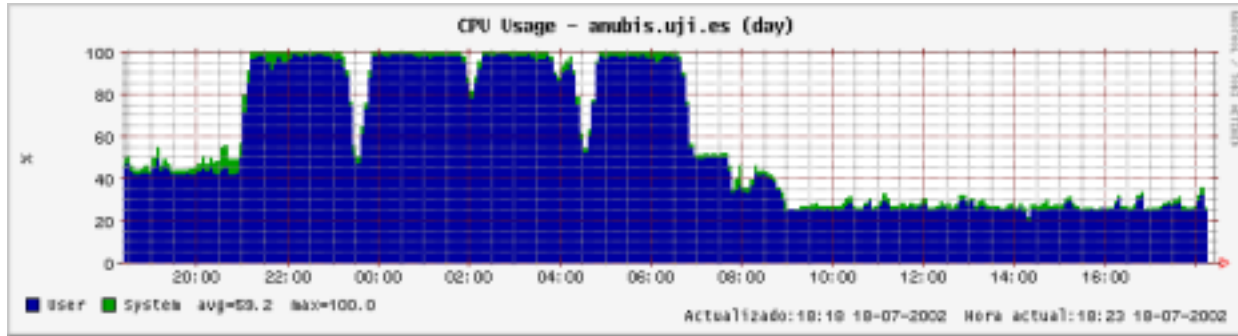


Figura 1. Consumo de CPU en anubis, un día

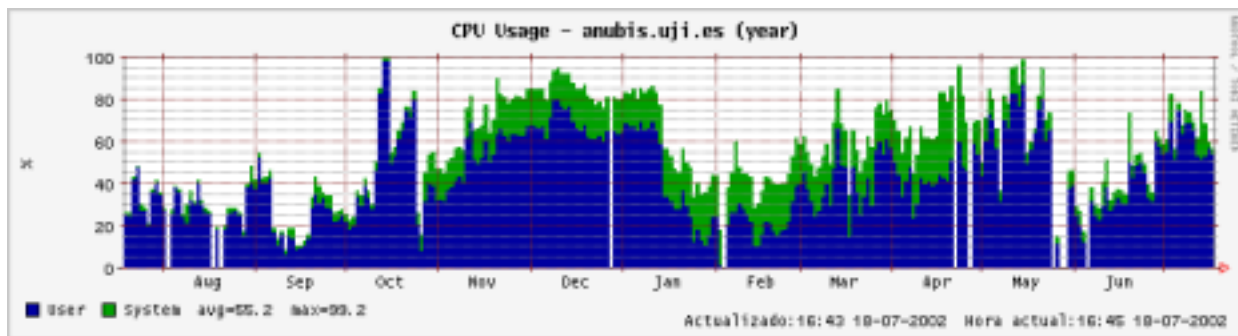


Figura 2. Consumo de CPU en anubis, un año

Se observa el tanto por cien de cpu usado, tanto por usuarios como por el sistema, al lo largo del periodo de tiempo representado. También se indica el máximo y la media de utilización durante ese periodo.

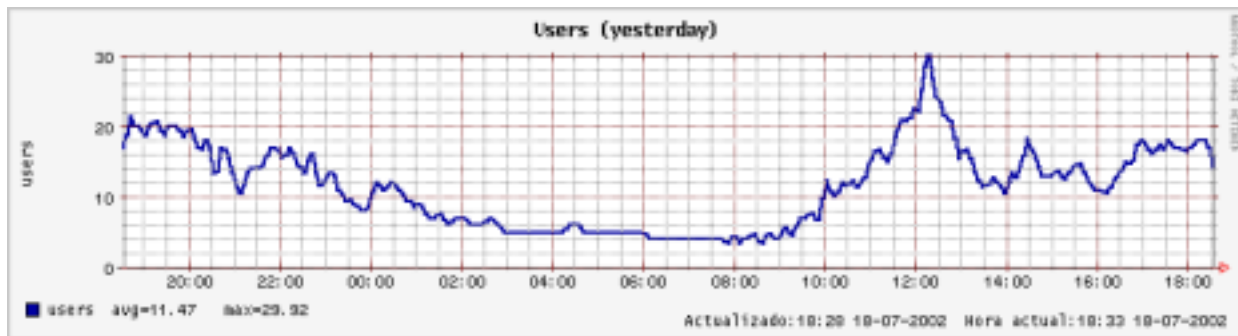


Figura 3. Número de usuarios.

Número de usuarios en el sistema. Este dato se obtiene a partir del comando `who`.

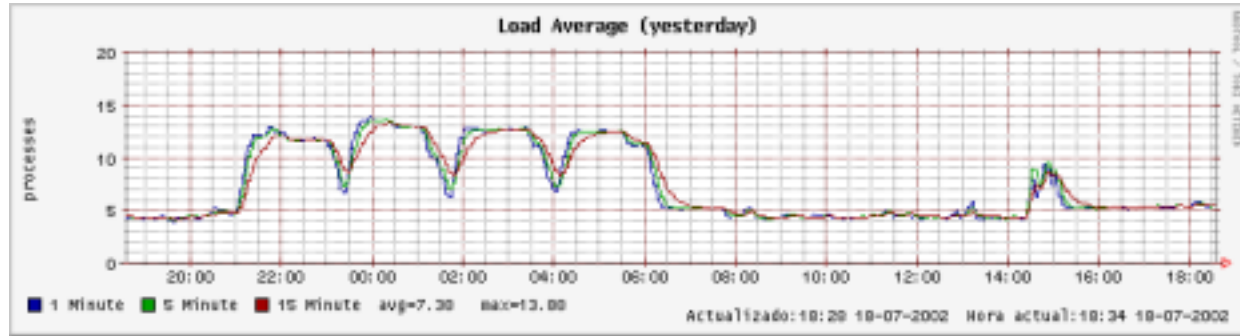


Figura 4. Carga del sistema. Número de procesos esperando CPU.

Cada una de las líneas representa la carga del sistema para el último minuto, cinco minutos y quince minutos respectivamente. Se obtiene a partir de la salida del comando `uptime`.

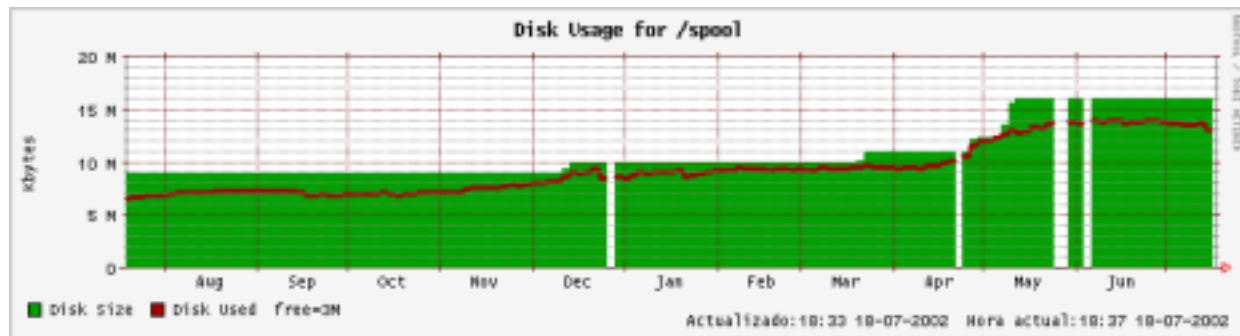


Figura 5. Ocupación de disco.

El área representa la capacidad del disco, mientras que la línea indica el disco que está siendo usado. Se obtiene de la salida del comando `df`.

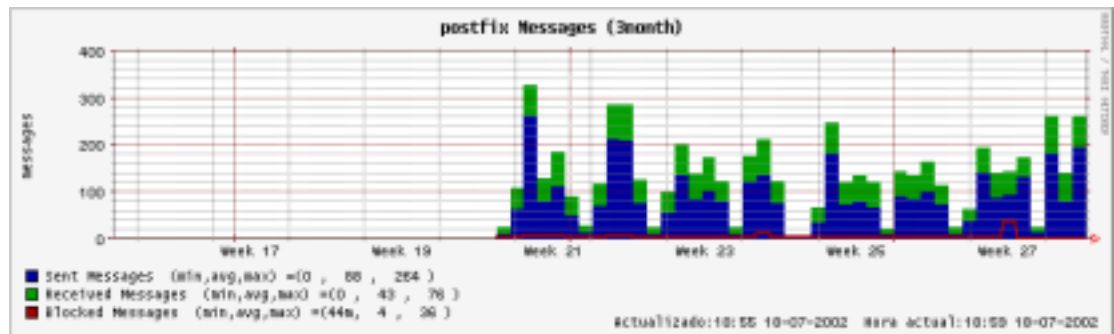


Figura 6. Mensajes de correo.

Muestra el número de mensajes de correo enviados, recibidos y bloqueados. Para ello se procesan las líneas de log del servidor de correo postfix.

RRDtool, herramienta de bases de datos en round robin.

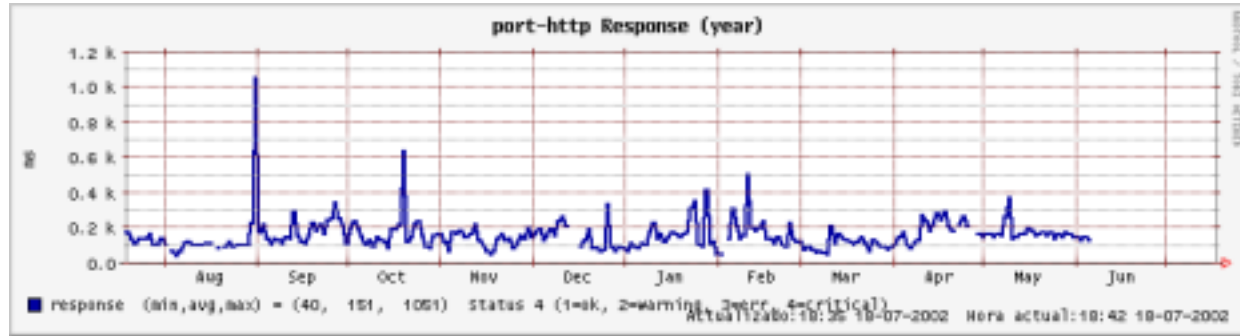


Figura 7. Tiempo de respuesta www.uji.es (antes de migración)

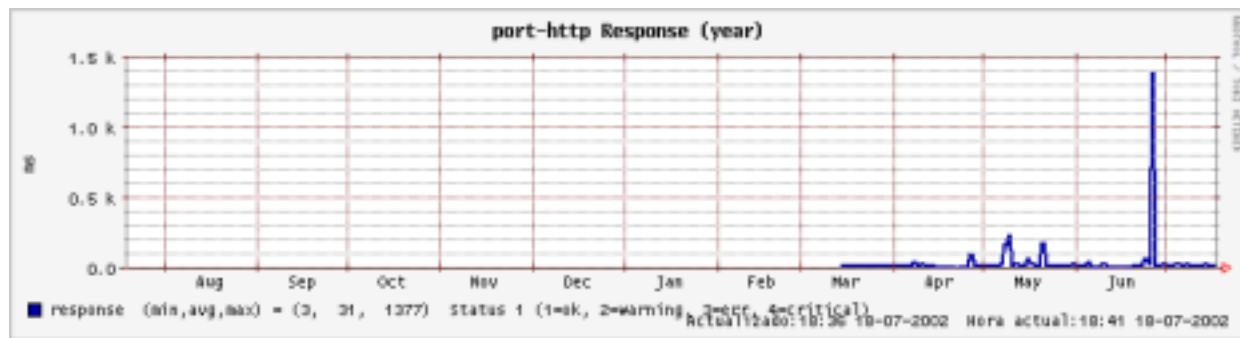


Figura 8. Tiempo de respuesta www.uji.es (después de migración)

Tiempo de respuesta del servidor web. Se obtiene a partir del tiempo que se tarda en obtener la página principal del servidor.

¿Cómo funciona internamente?

¿Cómo se almacenan los datos?

- se recogen con una frecuencia determinada
- se almacenan en diferentes "archivos", según su resolución y en paralelo

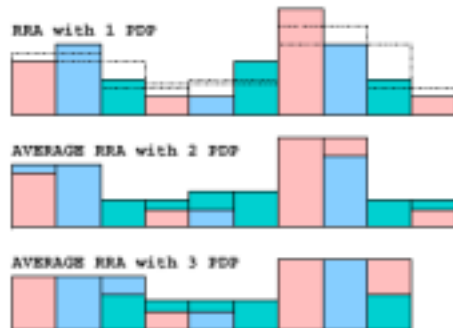


En esta base de datos RRD hay tres archivos RRA para diferentes periodos de tiempo. El archivo para 5 minutos contiene los últimos valores recogidos. El archivo para 1 hora contiene los valores para una hora, calculados a partir de 12 valores del anterior archivo RRA. Finalmente el archivo para un día, tiene un valor que resume la medida para un día. Este se obtiene a partir de 24 valores del RRA de 1 hora.

Cada uno de los archivos tiene un tamaño fijo, por lo que no crece en el tiempo. Existe un apuntador al último dato recogido. Los diferentes archivos se actualizan en paralelo a partir de las actualizaciones periódicas.

Funciones de consolidación.

- Ejemplo de cálculo de la media.



Los diferentes archivos se forman aplicando la función de consolidación a los valores originales.

Validación de los datos

- valor mínimo y máximo aceptable
- frecuencia mínima de actualización
- mínimo de valores en la consolidación

RRDtool, herramienta de bases de datos en round robin.

Cuando recogemos datos es importante que estos sean validos. De nada nos sirve recoger datos si no tienen ningún sentido. RRDtool nos permite establecer una serie de restricciones sobre los datos antes de procesarlos. Si no se cumple las restricciones esa muestra tomará el valor *UNKNOWN*.

Funciones de RRDtool

Funciones de RRDtool

- Interactuamos con la base de datos a través de las funciones de RRDtool
- Están ampliamente documentadas
- Tienen la misma sintaxis con independencia del lenguaje desde donde lo uses:
 - línea de comandos
 - módulo de perl
 - PHP
 - C

Sugerencia: Normalmente sólo usaras algunas de ellas, las más importantes son: *create*, *update*, *graph*, y la utilidad *rrdcgi*. En los siguientes puntos verás ejemplos de cómo se utilizan.

En la actualidad RRDtool soporta estas funciones:

- *create*
 - crea una nueva base de datos RRD
 - se crea con su tamaño final
 - se rellena con datos *UNKNOWN*
- *update*
 - almacena un nuevo conjunto de valores en el RRD
 - los datos se procesan según las propiedades del RRD
- *graph*
 - crea un gráfico a partir de los datos de uno o varios RRD
- *xport*
 - exportar datos en formato XML basados en uno o varios RRD
- *rrdcgi*
 - utilidad externa para producir gráficos bajo demanda
- *info*
 - extrae la información de definición del RRD
- *dump*
 - volcado de datos en XML

- *restore*
 - recuperación de datos con el formato del dump
- *fetch*
 - obtener datos de un determinado periodo de tiempo de un RRD la función `graph` se basa en ella
- *tune*
 - modifica la configuración de un RRD
- *last*
 - informa de la última actualización del RRD
- *rrdresize*
 - cambia el tamaño de un RRA, creando un nuevo RRD

Crear una base de datos RRD

A la hora de crear una base de datos RRD necesitamos saber una serie de datos acerca de lo que queremos medir. En concreto deberemos especificar la frecuencia de actualización, la naturaleza de los datos, las funciones que forman los archivos y la resolución de estos.

- Necesitamos saber:

la frecuencia de actualización

- debemos indicar cada cuanto tiempo esperamos servir datos a la base de datos
Ej. 300 segundos (5 minutos)

naturaleza de los datos

- *COUNTER*
 - contador, siempre se incrementa
 - registra el incremento/intervalo tiempo
 - Ej. tráfico de red, bytes/s
- *GAUGE*
 - indicador, distintos valores en el tiempo
 - registra el valor "tal como lo medimos" Ej. uso de cpu, número de usuarios, temperatura
- *DERIVE*
 - contador, puede decrecer
- *ABSOLUTE*
 - valor absoluto
 - contador que se resetea tras su lectura
 - registra el valor, el incremento es implícito

RRDtool, herramienta de bases de datos en round robin.

- Lo veremos mas claro con un ejemplo:
- Creamos la base de datos

```
rrdtool create all.rrd --start 978300900 \  
DS:a:COUNTER:600:U:U \  
DS:b:GAUGE:600:U:U \  
DS:c:DERIVE:600:U:U \  
DS:d:ABSOLUTE:600:U:U \  
RRA:AVERAGE:0.5:1:10
```

Nota: Definimos una variable de cada tipo.

- Insertamos los datos

```
rrdtool update all.rrd \  
978301200:300:1:600:300 \  
978301500:600:3:1200:600 \  
978301800:900:5:1800:900 \  
978302100:1200:3:2400:1200 \  
978302400:1500:1:2400:1500 \  
978302700:1800:2:1800:1800 \  
978303000:2100:4:0:2100 \  
978303300:2400:6:600:2400 \  
978303600:2700:4:600:2700 \  
978303900:3000:2:1200:3000
```

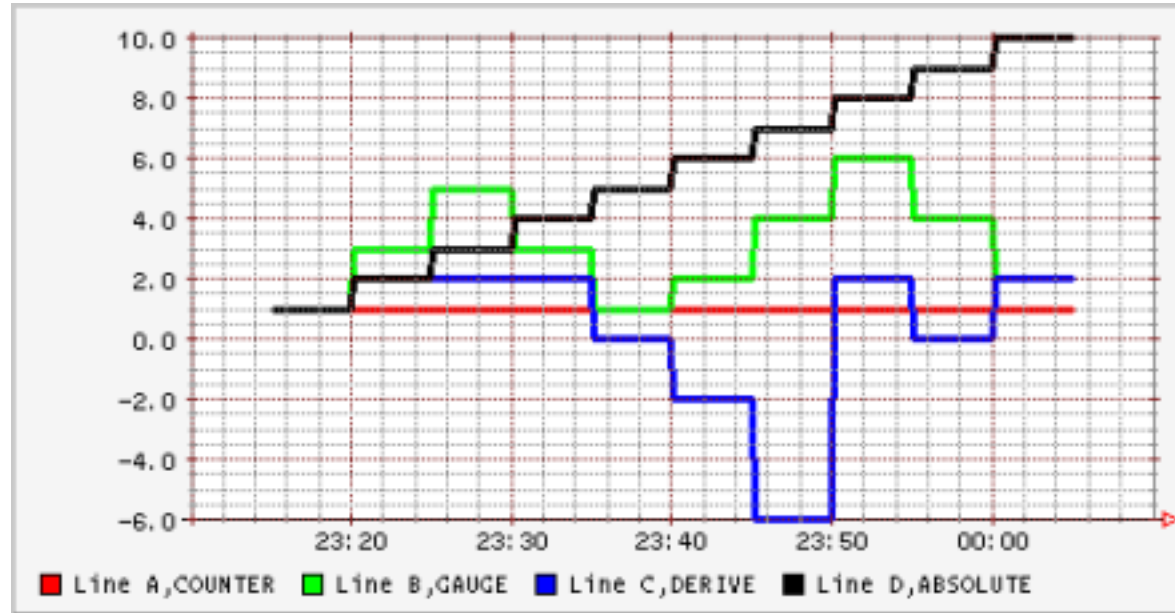
Nota: La primera columna indica la marca de tiempo, las siguientes son los valores para cada una de las cuatro variables.

- Creamos el gráfico

```
rrdtool graph all.png --start 978300600 --end 978304200 \  
-height 200 --imgformat PNG \  
DEF:linea=all.rrd:a:AVERAGE LINE3:linea#FF0000:"Line A,COUNTER" \  
DEF:lineb=all.rrd:b:AVERAGE LINE3:lineb#00FF00:"Line B,GAUGE" \  
DEF:linec=all.rrd:c:AVERAGE LINE3:linec#0000FF:"Line C,DERIVE" \  
DEF:lined=all.rrd:d:AVERAGE LINE3:lined#000000:"Line D,ABSOLUTE"
```

Nota: Pintamos una linea para cada variable, identificando el tipo en la leyenda del gráfico.

- Y este es el resultado:



Nota: Puedes observar que para la variable A, aunque insertamos datos que van aumentando, se dibuja una línea recta. Esta representa el incremento constante. La línea B, GAUGE, representa los valores tal como los recogemos. La línea C, DERIVE, se comporta como la A pero permite decremento. La línea D, ABSOLUTE, para los mismos datos que la A, representa un incremento mayor, ya que supone que se resetea el contador.

funciones de consolidación para los archivos RRA

- AVERAGE
- MIN
- MAX
- LAST

Indicará la función a aplicar sobre los datos originales para obtener valores de un RRA con menor resolución. Normalmente nos interesará guardarnos la media de los valores recogidos, pero también nos puede ser útil el mínimo o máximo valor del periodo.

resolución de los archivos RRA

- de cuantas muestras se compone cada RRA
- cuantos valores ha de guardar cada RRA

Especificamos la “resolución” de nuestro archivo y la “longitud” del mismo.

En resumen, una base de datos RRD se compone de:

- Data sources

Esta es la sintaxis para especificar una fuente de datos o data source (DS) en RRD:

DS: nombre: COUNTER | GAUGE | ABSOLUTE | DERIVE: frecuencia: vmin: vmax

RRDtool, herramienta de bases de datos en round robin.

- DS: clave para data source
 - *nombre*: variable donde recogeremos los datos
 - *tipo*: COUNTER, GAUGE, ABSOLUTE, DERIVE
 - *frecuencia*: especifica el número de segundos que pueden pasar entre cada muestra sin que se asuma como nulo.
 - *vmin*, *vmax*: valor mínimo y máximo aceptable. Si el valor recogido está fuera del rango, se le asignará el valor nulo.
- Round robin archives

Esta es la sintaxis para especificar un archivo de la base de datos, o round robin archive (RRA):

```
RRA:AVERAGE|MIN|MAX:porcentaje:muestras:filas
```

- RRA: clave para round robin archive
- *funcion*: función de consolidación a aplicar sobre los datos. AVERAGE, MIN, MAX, LAST
- *percent*: define la mínima cantidad de valores que deben ser conocidos para que el resultado no se nulo.
- *steps*: define la cantidad de valores originales se utilizan para calcular la función de consolidación.
- *rows*: especifica la cantidad de valores de este tipo que deben guardarse.

Esta será la sintaxis de la creación de la base de datos:

```
rrdtool create fichero.rrd
--start now-10s
--step 300
DS:nombre:COUNTER|GAUGE|DERIVE|ABSOLUTE:frecuencia:vmin:vmax
RRA:AVERAGE|MIN|MAX:porcentaje:muestras:filas
```

Un ejemplo, registro de temperaturas:

```
rrdtool create temp.rrd crea el fichero demo.rrd
--step 300 actualizable cada 5 min
DS:a:GAUGE:600:-30:50 guarda un dato "indicador" con frec max de 10 min
DS:b:GAUGE:600:-30:50 entre -30 y 50
RRA:AVERAGE:0.5:1:288 guarda la media de cada 5 min por un día
RRA:AVERAGE:0.5:12:168 la media de 1 hora por una semana
RRA:MIN:0.5:12:168 el mínimo de cada hora por una semana
RRA:MAX:0.5:12:168 el máximo de cada hora por una semana
```

Actualizar una base de datos RRD

- Utilizamos la función update.
- Se alimenta el RRD en una frecuencia "constante". Cada muestra se acompaña de una marca de tiempo, que puede ser el tiempo Unix o N para especificar el instante actual.
- La sintaxis es:

```
rrdtool update filename N|tiempo:valor[:valor]
```

- Actualizamos nuestra base de datos con temperaturas:

```
rrdupdate temp.rrd 1027034300:10:21
rrdupdate temp.rrd 1027034600:09:21
rrdupdate temp.rrd 1027034900:07:21
rrdupdate temp.rrd 1027035200:05:20
```

- Para especificar el momento actual:

```
rrdupdate temp.rrd N:16:20
```

Crear gráficos

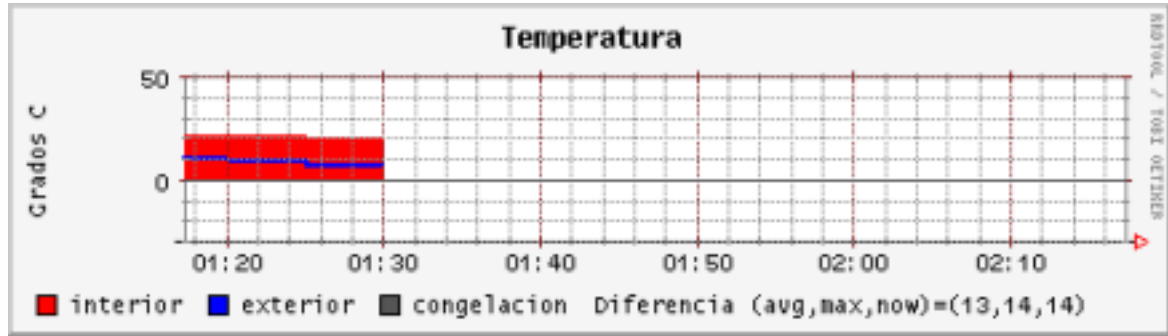
- Puedes crear gráficos sobre un RRD o utilizando variables de diferentes bases de datos.
- Puedes almacenar los datos de manera simple y realizar cálculos al representarlos gráficamente
- Utilizamos la función graph.

graph

- La sintaxis es muy extensa.
- Un ejemplo completito.

```
rrdtool graph temp.png
--imgformat PNG
--start now-lh      formato de tiempo at-style
--title "Temperatura" --vertical-label "Grados C"
--lower-limit -30 --upper-limit 50 --height 200
DEF:exterior=temp.rrd:a:AVERAGE
DEF:interior=temp.rrd:b:AVERAGE
CDEF:diff=interior,exterior,- cálculos dentro del gráfico
AREA:interior#ff0000:interior un área
LINE3:exterior#0000ff:exterior una línea
HRULE:0#555555:congelacion horizontal rule
GPRINT:diff:AVERAGE:"Diferencia (avg,max,now)=(%.01f,\g"
GPRINT:diff:MAX:"%.01f,\g"
GPRINT:diff:LAST:"%.01f)"
```

RRDtool, herramienta de bases de datos en round robin.



- También podemos generar gráficos con rrdcgi.

rrdcgi

- Se crean bajo demanda.
- Si especificamos la opción `--lazy` sólo se creará el gráfico si han cambiado los datos.
- Ejemplo:

```
#!/usr/bin/rrdcgi
<HTML>
<HEAD><TITLE>RRDCGI Demo</TITLE></HEAD>
<BODY>
<H1>RRDCGI Example Page</H1>
<P>
<RRD::GRAPH battery-hour.png --lazy --title="battery" --start now-
1h
    --title battery-hour
    --vertical-label "battery charge"
    DEF:percent=battery.rrd:charge:AVERAGE
    LINE2:percent#0000FF:"Battery charge">
</P>
</BODY>
</HTML>
```

Un ejemplo práctico.

Vamos a proponernos un problema típico:

- Se trata de registrar el tráfico de red de un determinado host.
- Usando línea de comandos.
- Utilizaremos como fuente de datos snmp

Sugerencia: si no lo conoces, mediante snmp podemos preguntar a equipos por los valores de ciertos contadores que mantiene

- Creamos la base de datos

```
#!/bin/bash
```

```
interface="eth0"
rrdtool create ${inter}.rrd \
  --step=300 \
  DS:input:COUNTER:600:0:U \
  DS:output:COUNTER:600:0:U \
  RRA:AVERAGE:0.5:1:288 \
  RRA:AVERAGE:0.5:12:168 \
  RRA:MAX:0.5:12:168 \
  RRA:MIN:0.5:12:168
```

- Preparamos un script para recoger los datos.

```
#!/bin/bash

ifInOctets="interfaces.ifTable.ifEntry.ifInOctets"
ifOutOctets="interfaces.ifTable.ifEntry.ifOutOctets"

while true
do
    eth0In=$(snmpget localhost public ${ifInOctets}.2)
    eth0Out=$(snmpget localhost public ${ifOutOctets}.2)

    #la salida del snmpget es:
    # "interfaces.ifTable.ifEntry.ifInOctets.2 = Counter32: 2463"
    # me quedo con el valor que viene detrás de los dos puntos

    rrdtool update eth0.rrd N:${eth0In/*:}:${eth0Out/*: }

    echo "$(date +%Y%m%d-%H:%M) -> \
        rrdupdate eth0.rrd $(date +%s):${eth0In/*:}:${eth0Out/*: }"

    sleep 300
done
```

- Estos son los datos que tenemos:

```
20020718-21:07 -> rrdupdate eth0.rrd 1027019245:532699726:2799737822
20020718-21:12 -> rrdupdate eth0.rrd 1027019546:533479106:2801336024
20020718-21:17 -> rrdupdate eth0.rrd 1027019846:536820828:2804247588
20020718-21:22 -> rrdupdate eth0.rrd 1027020147:550716100:2818138898
20020718-21:27 -> rrdupdate eth0.rrd 1027020448:553381980:2819224998
...
...
20020719-09:29 -> rrdupdate eth0.rrd 1027063767:1014211001:2470179285
20020719-09:34 -> rrdupdate eth0.rrd 1027064067:1016166403:2483886188
20020719-09:39 -> rrdupdate eth0.rrd 1027064368:1028025595:2492645833
```

- Esta sería una manera de generar los gráficos:

```
#!/bin/bash

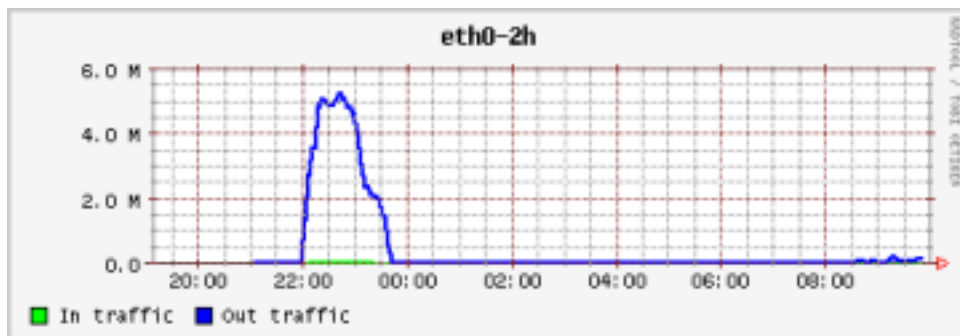
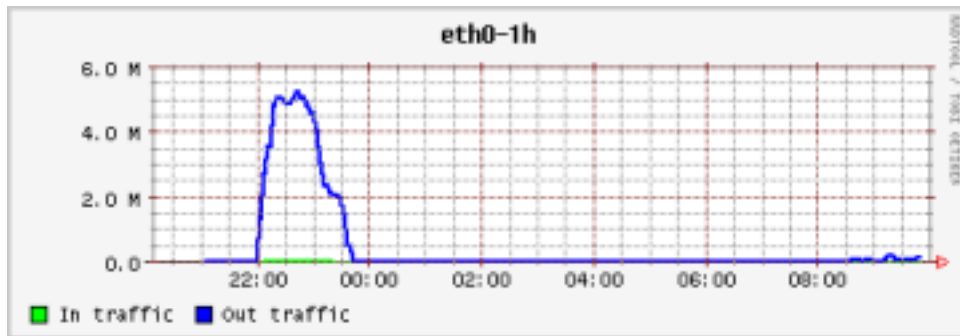
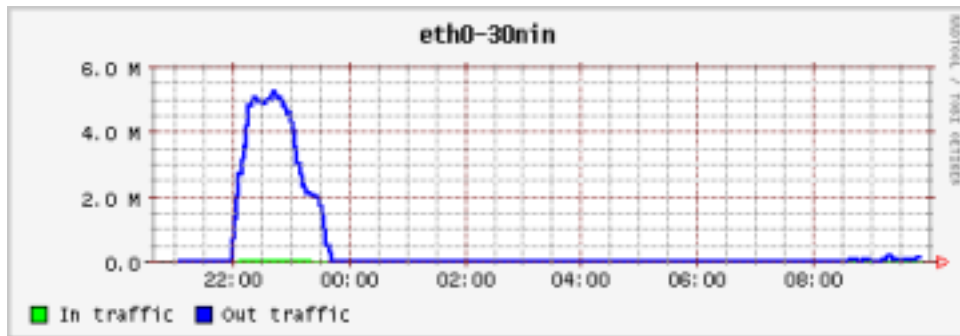
#start=now
#interfaces="eth0 ppp"
start=1027019245
interfaces="eth0"

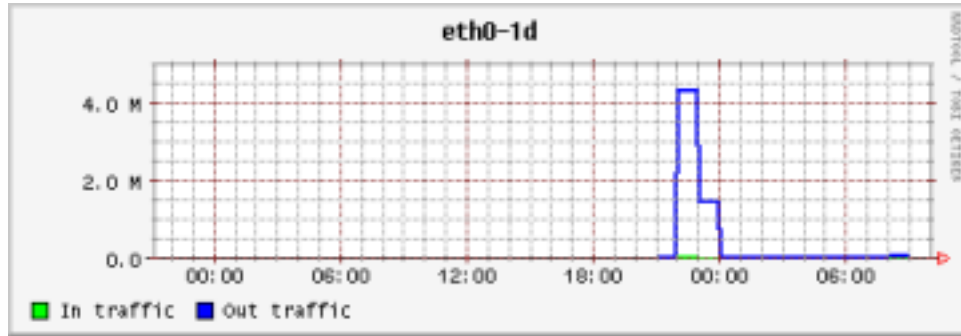
for rrd in ${interfaces}
do
    for past in 30min 1h 2h 1d
    do
        rrdtool graph ${rrd}-${past}.png --start ${start}-${past} \
            --imgformat PNG --title ${rrd}-${past} \
            DEF:inOctets=${rrd}.rrd:input:AVERAGE \
            DEF:outOctets=${rrd}.rrd:output:AVERAGE \
```

RRDtool, herramienta de bases de datos en round robin.

```
AREA:inoctets#00FF00:"In traffic" \
LINE1:outoctets#0000FF:"Out traffic"
done
done
```

- Y estos serían los gráficos:





Frontends: Remstats

Es uno de los varios frontends para RRDtool

- monitoriza remotamente un conjunto de servidores y routers
- recoge datos de rendimiento de manera centralizada
- almacena y mantiene los datos por largos espacios de tiempo
- produce gráficos agrupados en páginas web (*rrdcgi*)
- monitoriza estos datos para lanzar alertas

¿Cómo obtiene los datos?

- Se ejecuta un cron cada 5 minutos que se encarga de lanzar:
 - ping-collector
realiza ping desde el servidor central a los servidores monitorizados
 - port-collector
tiempo de respuesta y estado de un servicio (http,mail,etc). Abre conexión, dialoga con el protocolo interpretando el resultado.
 - snmp-collector
se interroga mediante snmp al equipo remoto
 - log-collector
se parsean líneas de log, extrayendo la información mediante expresiones regulares
 - unix-status-collector: **uname, uptime, vmstat, df, netstat, ps**
interpreta la salida de comandos en los servidores remotos a través de un servicio propio.
- Al recoger estos datos se analizan para poder lanzar alertas

Resultados

- Se puede comprobar a lo largo del tiempo el rendimiento de diversos servidores
- Se reciben alertas cuando el sistema necesita intervención
- Se pueden comparar datos fácilmente entre diferentes servidores
- Se pueden comparar datos del mismo equipo
router: tráfico, consumo de CPU, temperatura

Conclusiones

Con RRDtool podemos:

- almacenar datos de una manera compacta
- crear gráficos sobre los datos fácilmente

Se aplica directamente a:

- datos que varían a través del tiempo
- sobre los que queremos observar evolución
- útil para observar el rendimiento de servidores
- a partir de comandos del sistema: **uptime, vmstat,df, netstat, ps, /proc**
- o consultas snmp

Junto con alguno de los frontends:

- se puede crear un sistema de monitorización
- nos permite observar el estado de nuestros servidores y equipos de comunicaciones
- las alarmas indican posibles problemas

Más información

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

- fuentes
- binarios
- amplia documentación
- tutoriales
- enlaces a frontends

Remstats <http://remstats.sourceforge.net/release/index.html> ²

- fuentes
- fuentes
- documentación
- live data

RRGrapher <http://net.doit.wisc.edu/~plonka/RRGrapher/> ³

- CGI en Perl que te permite crear gráficos sobre los RRD de tu sistema.

Otros usos de RRDtool:

- Monitoring Natural Gas Usage <http://phk.freebsd.dk/Gasdims/> ⁴
- Una carrera de karts <http://faq.mrtg.org/example/karting.html> ⁵

Notas

1. <http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/>
2. <http://remstats.sourceforge.net/release/index.html>
3. <http://net.doit.wisc.edu/~plonka/RRGrapher/>
4. <http://phk.freebsd.dk/Gasdims/>
5. <http://faq.mrtg.org/example/karting.html>

RRDtool, herramienta de bases de datos en round robin.