Monitoreo de redes con Munin

Gunnar Wolf — gwolf@debian.org
http://www.gwolf.org/seguridad/munin

Instituto de Investigaciones Económicas, UNAM
Desarrollador del proyecto Debian

2º Foro Nacional de Software Libre FONASOL 2008
28-30 de mayo, Coatzacoalcos, Veracruz

Contents

¿Qué hace un administrador de red/sistemas?

- Asegurar la disponibilidad de los servicios
- Asegurar la confianza en la información provista/almacenada
- Conocer en todo momento el estado de sus equipos, para poder anticiparse a los problemas
- ...Y, claro, mucho más

Munin es una infraestructura genérica de monitoreo histórico y graficación de servicios/recursos

¿Monitoreo histórico?

Una de las tareas más importantes que, como administradores de sistemas o de redes, tenemos la responsabilidad de realizar periódicamente, para encontrar tendencias y anticiparse a los problemas, es el

Monitoreo de recursos

¿Qué significa? ¿Por qué tengo que hacerlo? ¿De qué me sirve? ¿Y cómo puedo hacerlo divertido?

¿Divertido?

¿Dije acaso *Divertido*?

¿No se supone que mi trabajo es mi *obligación*, y que hasta me pagan por ello?
Es cierto, pero...

¿Divertido?

¿Dije acaso *Divertido*?

¿No se supone que mi trabajo es mi *obligación*, y que hasta me pagan por ello?
Es cierto, pero...

A fin de cuentas, lo divertido es muy subjetivo.
¡Y no por ser divertido tiene que ser menos serio!

¿Divertido?

¿Dije acaso *Divertido*?

¿No se supone que mi trabajo es mi *obligación*, y que hasta me pagan por ello?
Es cierto, pero...

A fin de cuentas, lo divertido es muy subjetivo.
¡Y no por ser divertido tiene que ser menos serio!

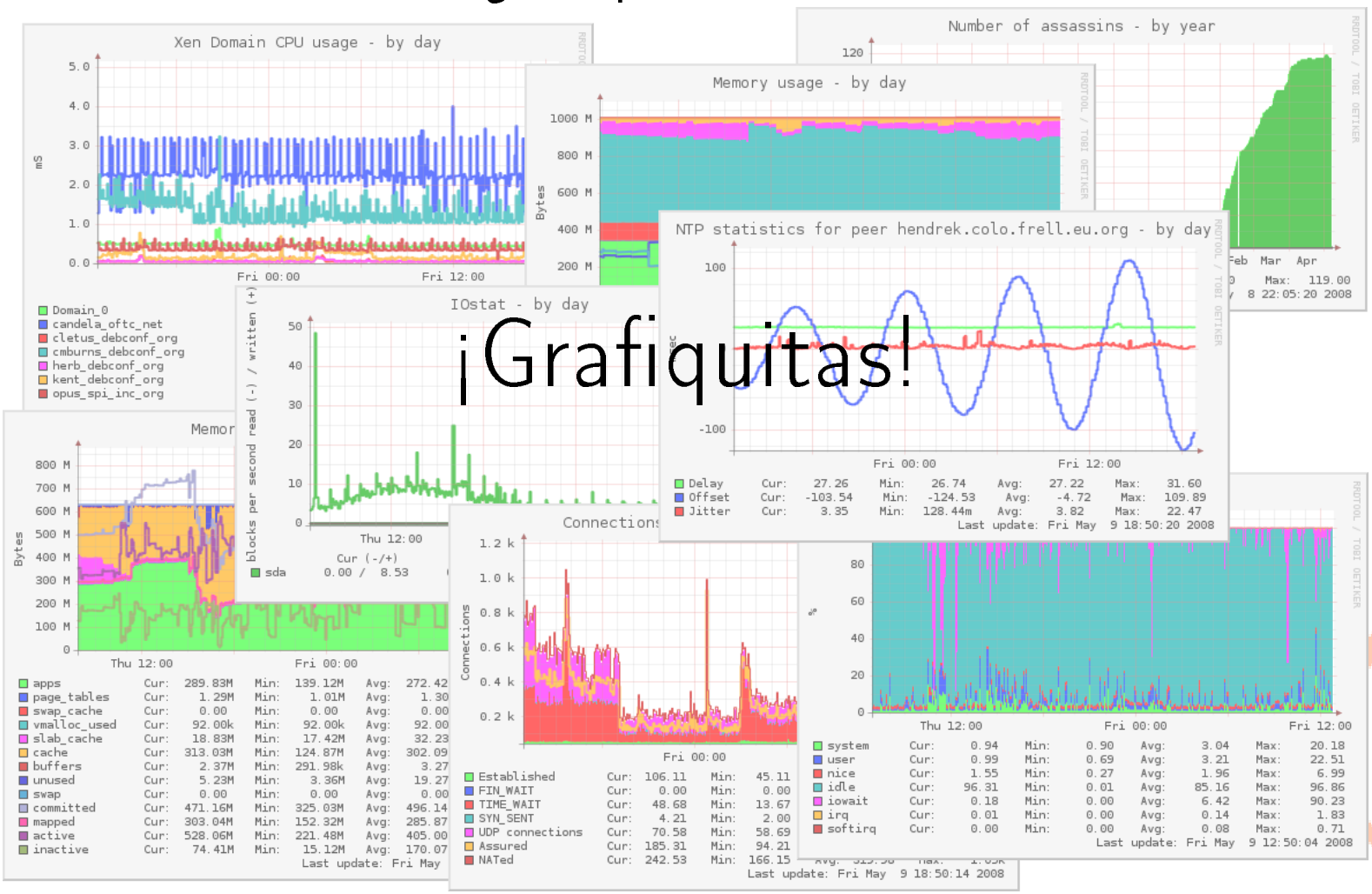
Total... ¿Negarán acaso que somos una bola de *nerds/geeks* bastante atípicos y que tenemos un concepto muy enfermo de la diversión?

¿Divertido? ¡Grafiquitas! :-D

Pero... ¿Qué puede ser *divertido*?

¿Divertido? ¡Grafiquitas! :-D

Pero... ¿Qué puede ser *divertido*?



¿Divertido? ¡Grafiquitas! :-D

Pero... ¿Qué puede ser *divertido*?

¿Divertido? ¡Grafiquitas! :-D

Pero... ¿Qué puede ser *divertido*?

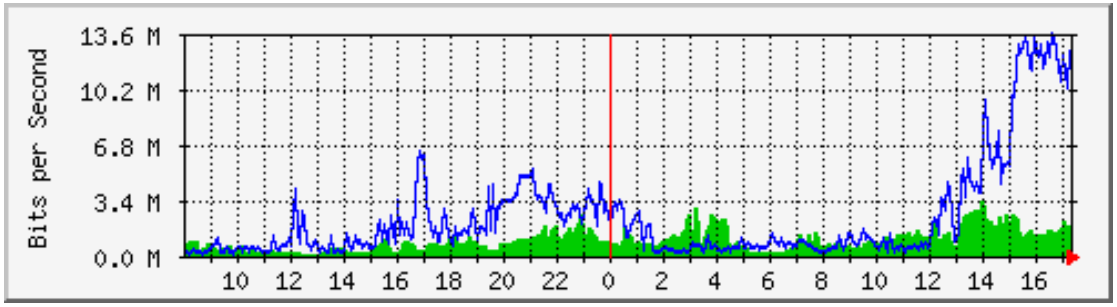
El verdadero valor de las gráficas

Hablando completamente en serio...

- Las gráficas históricas son una herramienta fundamental en el arsenal de un administrador de sistemas
- El que conserven memoria histórica es fundamental para analizar tendencias y encontrar comportamientos a largo plazo
- Entre menos tengamos que configurar/batallar para lograr una primer imagen satisfactoria del sistema de monitoreo, más vamos a animarnos a utilizarlo
- Entre más podamos adecuar la información que nos es presentada a las necesidades específicas de nuestro entorno, más útil nos va a ser el sistema que elijamos
- **Nos ayudan muchísimo a aprender**, a comprender el funcionamiento y relaciones dentro de nuestro sistema

Contents

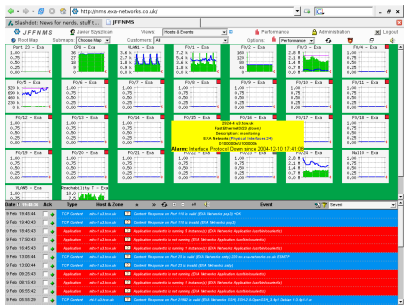
MRTG — Multi Router Traffic Grapher

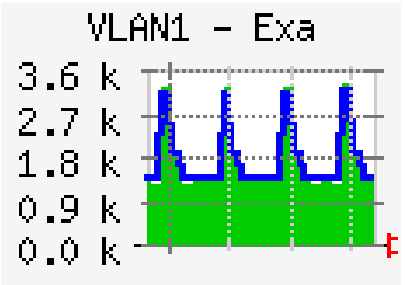


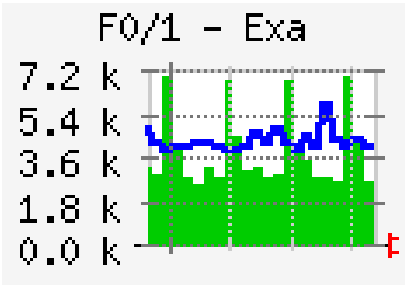
http://oss.oetiker.ch/mrtg/

- Creado en 1994, probablemente el pionero en este campo
- Orientado a mostrar el rendimiento de una interfaz de red, corriendo directamente en el servidor, o vía SNMP a un ruteador
- Relativamente extensible — Pero limitado a mostrar dos valores por gráfica (tráfico entrante y saliente)

JFFNMS — Just For Fun Network Management System



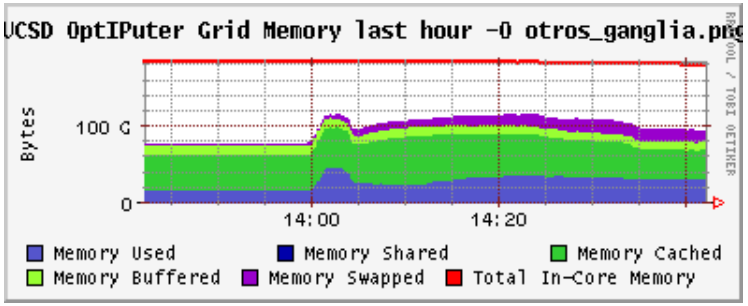




`http://jffnms.sourceforge.net/`

- Monitoreo orientado a SNMP, Syslog, Tacacs+
- modular, extensible
- Tiene funcionalidad de autodescubrimiento de dispositivos/recursos en red
- interfaz mostrando resumenes del estado de la red muy bien desarrolladas
- Generación de reportes dinámica (PHP)

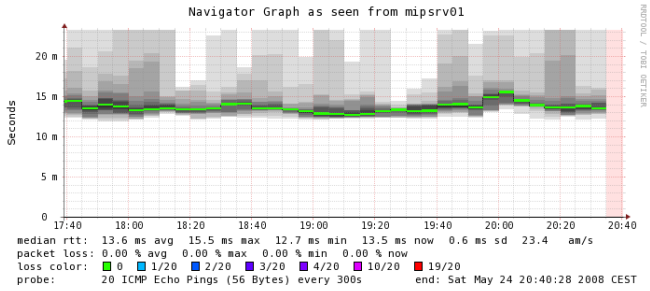
Ganglia



http://ganglia.info/

- Creado por la Universidad de California en Berkeley para el *Millenium Project* (http://www.millennium.berkeley.edu/)
- Enfocado al monitoreo de entornos de cómputo distribuído de alto rendimiento (clusters y grids)
- Actualmente se utiliza para monitorear clusters de hasta 2000 nodos en diversas universidades de todo el mundo

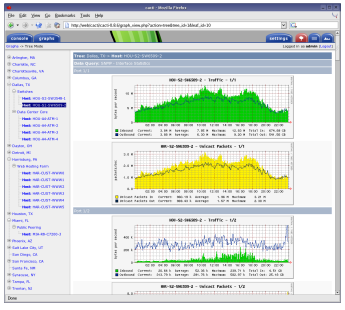
Smokeping

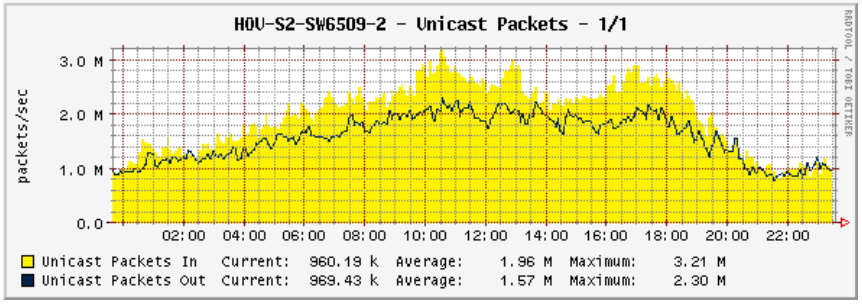


`http://oss.oetiker.ch/smokeping/`

- Orientado a monitorear la *latencia* en nuestra red
- Al estar tan especializado, incluye un sistema de filtrado que sólo nos muestra las gráficas *interesantes*
- Tiene modos de operación maestro/esclavo, en la que un host maestro puede indicar a todos los que monitorea hacia qué equipo dirigir sus pings
- Muestra la desviación estándar, presentándonos no sólo qué tan cargada está en determinado momento nuestra red, sino también *qué tan normal* es este patrón

Cacti





http://www.cacti.net/

- Framework de monitoreo completamente configurado por Web, hecho en PHP
- Principalmente orientado a monitorear dispositivos SNMP; tiene capacidad de autodescubrimiento de recursos para dichos equipos
- Viene con varios templates predefinidos para varias clases de dispositivos no-SNMP (p.ej. hosts Unix)
- Interfaz amigable y potente, permite acercarse a determinados intervalos interactivamente (cosa que pocos sistemas ofrecen)

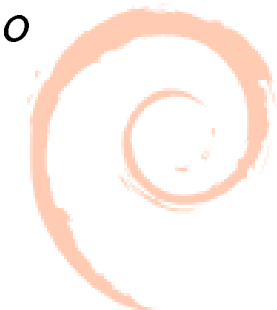
RRDtool

Se habrán dado cuenta que todos estos sistemas tienen algo en común... Las gráficas de todos son sospechosamente parecidas.

Todos estos sistemas están construídos en torno a *RRDtool*, `http://oss.oetiker.ch/rrdtool/` — Escrito por Tobi Oetiker (también autor de MRTG y Smokeping)

RRDtool

- Round Robin Database — Una base de datos creada para guardar datos numéricos orientados a este tipo de gráficas
- Originalmente, RRDtool era parte de MRTG; fue separado para convertirse en un proyecto por derecho propio.
- Al ser creado un archivo de RRDtool, es creado con los parámetros de uso esperado que tendrá: El tiempo total a guardar, con la resolución con la que lo requeriremos en el futuro
- Una importante particularidad de las bases RRD es que *no crecen con el tiempo*.



RRDtool

- RRDTool tiene *bindings* para ser usado desde Perl, Python, Ruby, TCL y PHP
- Además de guardar los datos, claro, implementa la generación de sus gráficas
- RRDtool es una bestia muy interesante, vale la pena aprenderlo — Al menos, comprender sus conceptos básicos nos va a ayudar si (como veremos más adelante) escribimos plugins para Munin
- No vamos a profundizar sobre el tema. Hay un tutorial sencillo, completo y bien escrito (¡y en español!) al respecto en Bulma: `http://bulma.net/body.phtml?nIdNoticia=1284`

Contents

Munin y Hugin

Hugin y Munin en los hombros de Odín;

manuscrito islandés, s. XVIII

Huginn ok Muninn fljúga hverjan dag Jörmungrund yfir
óumk ek of Hugin, at hann aftr né komi⸗t, thó sjámk meir of Munin.

Hugin y Munin vuelan todos los días alrededor del mundo;
Temo menos por Hugin de que no regrese, aún más temo por Munin

Edda poética - Grímnismál, estrofas 19 y 20

Munin y Sugin

Representación de Munin y Hugin

Odín en el detalle de un casco, acompañado de sus cuervos.

En la mitología nórdica, Munin y Hugin son los cuervos del dios Odín. Vuelan a través del mundo, y relatan a Odín, susurrando a sus oídos, lo que han visto, todas las noticias.

Recuerdan todo.

Hugin es el *pensamiento* y Munin es la *memoria*.

Es debido a estos cuervos que el kennigar (Hrafnaguð) «dios cuervo» se utilizaba para referirse a Odín.

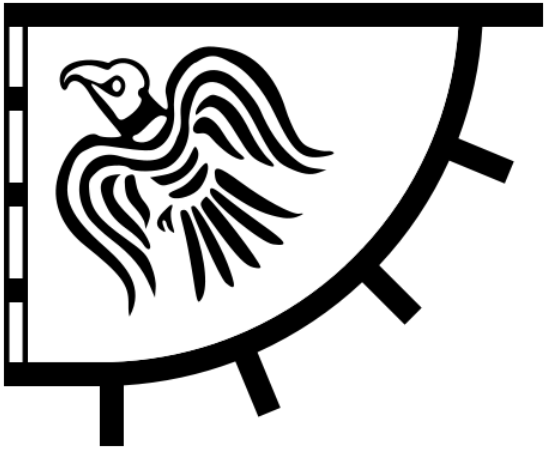
Munin y Sugin

Sugin y Munin coinciden con los dos hemisferios del cerebro y sus dos funciones: Pensar y recordar.

Sonaja en forma de cuervo

En la cultura Céltica el Cuervo es animal de presagios, de hecho está asociado a la brujería en toda Europa. Interpretando la voz de los cuervos puedes oír a los dioses.

Munin y Sugin



Estandarte vikingo del cuervo

Odín y Slepnir, el caballo de ocho patas

Cuando un cuervo planea encima de un guerrero significa buena suerte en la batalla.
Se dice que el emperador Federico Barba Roja está enterrado en el monte Kyffhäuse. Si algún día ningún cuervo planea más sobre este monte, el emperador regresara con sus huestes.
Esta saga rememora a Odin y sus *Einherjer*, los espíritus de los guerreros muertos en batalla preparándose para la batalla final.

Munin y Hugin — Referencias

- http://altreligion.about.com/
- http://web.telia.com/~u85903393/hugin_och_munin_english.html
- http://es.wikipedia.org/wiki/Hugin_y_Munin
- http://es.wikipedia.org/wiki/Estandarte_del_cuervo
- Y obviamente, http://munin.projects.linpro.no/

Munin y Hugin — Referencias

- http://altreligion.about.com/
- http://web.telia.com/~u85903393/hugin_och_munin_english.html
- http://es.wikipedia.org/wiki/Hugin_y_Munin
- http://es.wikipedia.org/wiki/Estandarte_del_cuervo
- Y obviamente, http://munin.projects.linpro.no/

Contents

¿Qué me llevó a Munin?

Habiendo tantos programas de dónde elegir.
¿Por qué Munin?

- Fácil de configurar e integrar, tanto en un sólo host como en una red, plana o compleja
- Orientado a la autoconfiguración (`munin-node-configure`)
- Diseñado para ser simple de extender a través de plugins
- Presenta información histórica que me permite analizar tendencias, pero *no* permite a un usuario remoto ejecutar código (genera HTML estático)

¿Qué **no es** Munin?

Munin **no es perfecto** ni es para todas las situaciones y configuraciones

- Munin recibe la información *sin autenticación* y *en texto plano* sobre la red. Si hay información sensible o confidencial, no es adecuado
- Hace monitoreo *periódico*, cada cinco minutos. Sirve para recolectar datos estadísticos, *no como herramienta de alertamiento* (puede servir, pero no es su función natural ni óptima)

Servidor, plugins y cliente

Munin se divide en tres componentes principales:

Servidor Un demonio que corre en todas las máquinas monitoreadas, por default en el puerto 4949. Su función es configurar y llamar a los plugins. Cuando hablamos de `munin-node`, nos referimos al servidor.

Plugins Cada uno de los agentes de recolección de datos que son invocados por `munin-node`. Dan la información que monitorean, y son también capaces de *describir su función y configuración*

Cliente Proceso que corre periódicamente (normalmente cada 5 minutos) desde un nodo central, interrogando a cada uno de los servidores `munin-node`, y generando las páginas Web con los resultados

Instalación de Munin

Puedo asumir que utilizan software de calidad, que no les gusta complicarse la vida...
¿Verdad?

Instalación de Munin

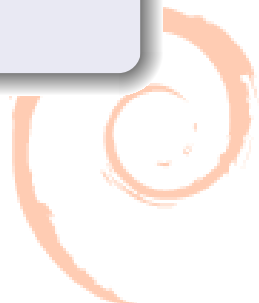
Puedo asumir que utilizan software de calidad, que no les gusta complicarse la vida...
¿Verdad?

Instalando Munin en Debian

```
gwolf@malenkaya[1] $ su -
Password:
root@malenkaya[1] # aptitude install munin munin-node
(...)
Do you want to continue? [Y/n/?] y
(...)
```

Y ya.

Instalado, configurado y funcionando.



Explicando la magia: munin-node-config

Un buen mago jamás revela sus secretos.

Explicando la magia: munin-node-config

Un buen mago jamás revela sus secretos.
Afortunadamente, soy administrador de sistemas, no mago.

munin-node-configure

Llamado sin opciones, nos desglosa la configuración actual:

```
0 gwolf@malenkaya[1]~$ /usr/sbin/munin-node-configure
Plugin                  | Used | Extra information
------                  | ---- | -----------------
acpi                    | yes  |
apache_accesses         | no   |
apache_processes        | no   |
apache_volume           | no   |
apt                     | no   |
apt_all                 | no   |
courier_mta_mailqueue   | no   |
courier_mta_mailstats   | no   |
courier_mta_mailvolume  | no   |
                        (...)
hddtemp_smartctl        | no   |
if_                     | yes  | eth1 eth2
if_err_                 | no   |
interrupts              | yes  |
```

Explicando la magia: munin-node-config

Indicando -suggest, nos *sugiere* qué plugins activar y con qué configuraciones lanzarlos

munin-node-configure -suggest

```
0 gwolf@malenkaya[2]~$ /usr/sbin/munin-node-configure --suggest
Plugin                  | Used | Suggestions
------                  | ---- | -----------
apache_accesses         | no   | [no apache server-status or ExtendedStatus missing on ports 80]
apache_processes        | no   | [no apache server-status on ports 80]
apache_volume           | no   | [no apache server-status or ExtendedStatus missing on ports 80]
courier_mta_mailqueue   | no   | [spooldir not found]
courier_mta_mailstats   | no   | [could not find executable]
courier_mta_mailvolume  | no   | [could not find executable]
cupsys_pages            | no   | [logfile not readable]
exim_mailqueue          | no   | [exim not found]
exim_mailstats          | no   |
hddtemp_smartctl        | no   | [smartctl not found]
if_                     | yes  | -eth2
if_err_                 | no   | yes +eth1
```

Claro, al momento de instalación, el sistema es interrogado por este script — Y obtenemos una configuración por omisión muy completa.

Configurando Munin-node (servidor) más allá de los defaults

¿Y si quiero afinar lo que me es monitoreado? ¿Agregar, quitar agentes? ¿Especificar parámetros?
Los parámetros base de cada servidor `munin-node` son configurados en /etc/munin/munin-node.conf:

```
log_level 4
log_file /var/log/munin/munin-node.log
port 4949
pid_file /var/run/munin/munin-node.pid
background 1
setseid 1

# Which port to bind to;
host 127.0.0.1
user root
group root
setsid yes
```

Para un sólo nodo, la configuración default es típicamente adecuada.

Configurando Munin-node (servidor) más allá de los defaults

Para determinar los plugins a activar, basta ligarlos o eliminarlos de /etc/munin/plugins:

```
lrwxrwxrwx 1 root root 29 2008-03-22 16:06 acpi -> /usr/share/munin/plugins/acpi
lrwxrwxrwx 1 root root 28 2008-03-22 16:06 cpu -> /usr/share/munin/plugins/cpu
lrwxrwxrwx 1 root root 27 2008-03-22 16:06 df -> /usr/share/munin/plugins/df
lrwxrwxrwx 1 root root 33 2008-03-22 16:06 df_inode -> /usr/share/munin/plugins/df_inode
lrwxrwxrwx 1 root root 32 2008-03-22 16:06 entropy -> /usr/share/munin/plugins/entropy
lrwxrwxrwx 1 root root 30 2008-03-22 16:06 forks -> /usr/share/munin/plugins/forks
lrwxrwxrwx 1 root root 28 2008-03-22 16:06 if_eth1 -> /usr/share/munin/plugins/if_
lrwxrwxrwx 1 root root 28 2008-04-05 15:21 if_eth2 -> /usr/share/munin/plugins/if_
(...)
```

Tomen nota de los últimos dos — Los plugins cuyo nombre termina en _ son *mágicos*

Configurando Munin-node (servidor) más allá de los defaults

- Algunos plugins pueden recibir, además, parámetros o configuraciones adicionales.
- Estos los configuramos a través de archivos en `/etc/munin/plugin-conf.d/`, en bloques con el estilo general/tradicional `.ini`
- Hay tres tipos de configuración en este archivo:

  user, group Especifica con qué usuario y grupo correr el plugin

  command Comando a ejecutar en vez del plugin. Podemos pasarle el *nombre* del plugin con %c

  env.* Variables de ambiente a configurar, para especificar parámetros particulares a cada plugin

- ...En una instalación básica, rara vez hace falta siquiera tocar estos archivos

Configurando Munin (cliente) más allá de los defaults

¿Y cómo monitoreo a varios hosts?
La configuración del cliente esta en /etc/munin/munin.conf; la sección relevante default dice:

```
[localhost.localdomain]
    address 127.0.0.1
    use_node_name yes
```

Configurando Munin (cliente) más allá de los defaults

Para monitorear a varios hosts, es casi igual:

```
[webserver.localdomain]
    address 127.0.0.1
    use_node_name yes

[database.localdomain]
    address 192.168.100.150
    use_node_name yes
```

Hay varias opciones más disponibles para la generación de páginas de gráficas (p.ej. totalizaciones, ordenamiento interno...)

El protocolo básico de Munin

El servidor de munin implementa un protocolo muy sencillo:

list	Muestra los plugins disponibles en este host
nodes	Nodos que este host reporta (p.ej. sobre SNMP)
config *plugin*	Descripción y configuración del plugin especificado
fetch *plugin*	Recupera los valores actuales del plugin solicitado
quit	Finaliza la sesión de monitoreo

Una conversación ejemplo con Munin-node (1)

```
$ nc localhost 4949
# munin node at webserver.localdomain
help
# Unknown command. Try list, nodes, config, fetch,
version or quit
list
open_inodes entropy irqstats mysql_slowqueries
processes acpi mysql_threads df netstat interrupts
swap mysql_bytes if_eth2 load df_inode cpu if_eth1
mysql_queries forks iostat open_files memory vmstat
nodes
webserver.localdomain
.
```

Una conversación ejemplo con Munin-node (1)

```
$ nc localhost 4949
# munin node at webserver.localdomain
help
# Unknown command. Try list, nodes, config, fetch,
version or quit
list
open_inodes entropy irqstats mysql_slowqueries
processes acpi mysql_threads df netstat interrupts
swap mysql_bytes if_eth2 load df_inode cpu if_eth1
mysql_queries forks iostat open_files memory vmstat
nodes
webserver.localdomain
.
```

Una conversación ejemplo con Munin-node (1)

```
$ nc localhost 4949
# munin node at webserver.localdomain
help
# Unknown command. Try list, nodes, config, fetch,
version or quit
list
open_inodes entropy irqstats mysql_slowqueries
processes acpi mysql_threads df netstat interrupts
swap mysql_bytes if_eth2 load df_inode cpu if_eth1
mysql_queries forks iostat open_files memory vmstat
nodes
webserver.localdomain
.
```

Una conversación ejemplo con Munin-node (1)

```
$ nc localhost 4949
# munin node at webserver.localdomain
help
# Unknown command. Try list, nodes, config, fetch,
version or quit
list
open_inodes entropy irqstats mysql_slowqueries
processes acpi mysql_threads df netstat interrupts
swap mysql_bytes if_eth2 load df_inode cpu if_eth1
mysql_queries forks iostat open_files memory vmstat
nodes
webserver.localdomain
.
```

Una conversación ejemplo con Munin-node (2)

```
config entropy
graph_title Available entropy
graph_args -base 1000 -l 0
graph_vlabel entropy (bytes)
graph_scale no
graph_category system
graph_info This graph shows the amount of entropy
available in the system.
entropy.label entropy
entropy.info The number of random bytes available.
This is typically used by cryptographic applications.
.
fetch entropy
entropy.value 815
.
quit
```

Una conversación ejemplo con Munin-node (2)

```
config entropy
graph_title Available entropy
graph_args -base 1000 -l 0
graph_vlabel entropy (bytes)
graph_scale no
graph_category system
graph_info This graph shows the amount of entropy
available in the system.
entropy.label entropy
entropy.info The number of random bytes available.
This is typically used by cryptographic applications.
.
fetch entropy
entropy.value 815
.
quit
```

Una conversación ejemplo con Munin-node (2)

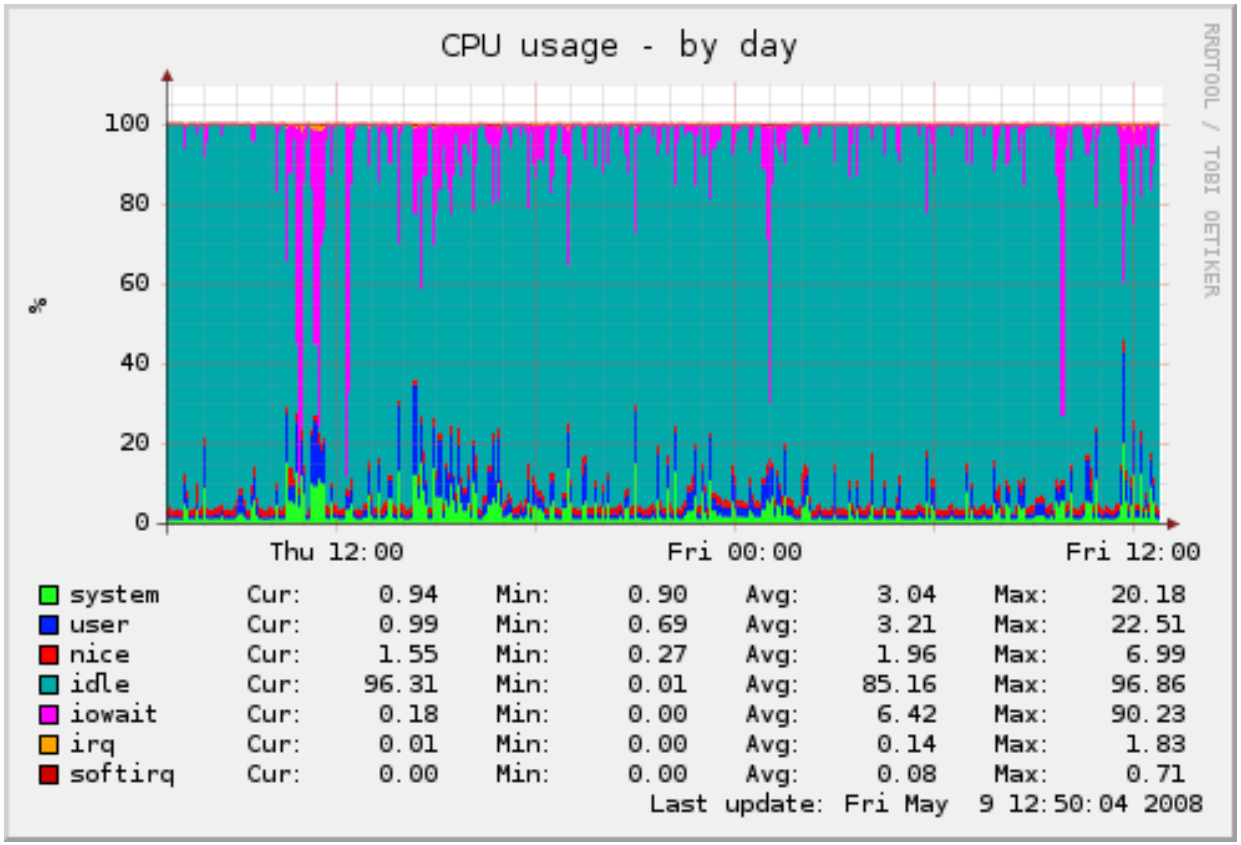
```
config entropy
graph_title Available entropy
graph_args -base 1000 -l 0
graph_vlabel entropy (bytes)
graph_scale no
graph_category system
graph_info This graph shows the amount of entropy
available in the system.
entropy.label entropy
entropy.info The number of random bytes available.
This is typically used by cryptographic applications.
.
fetch entropy
entropy.value 815
.
quit
```

Contents

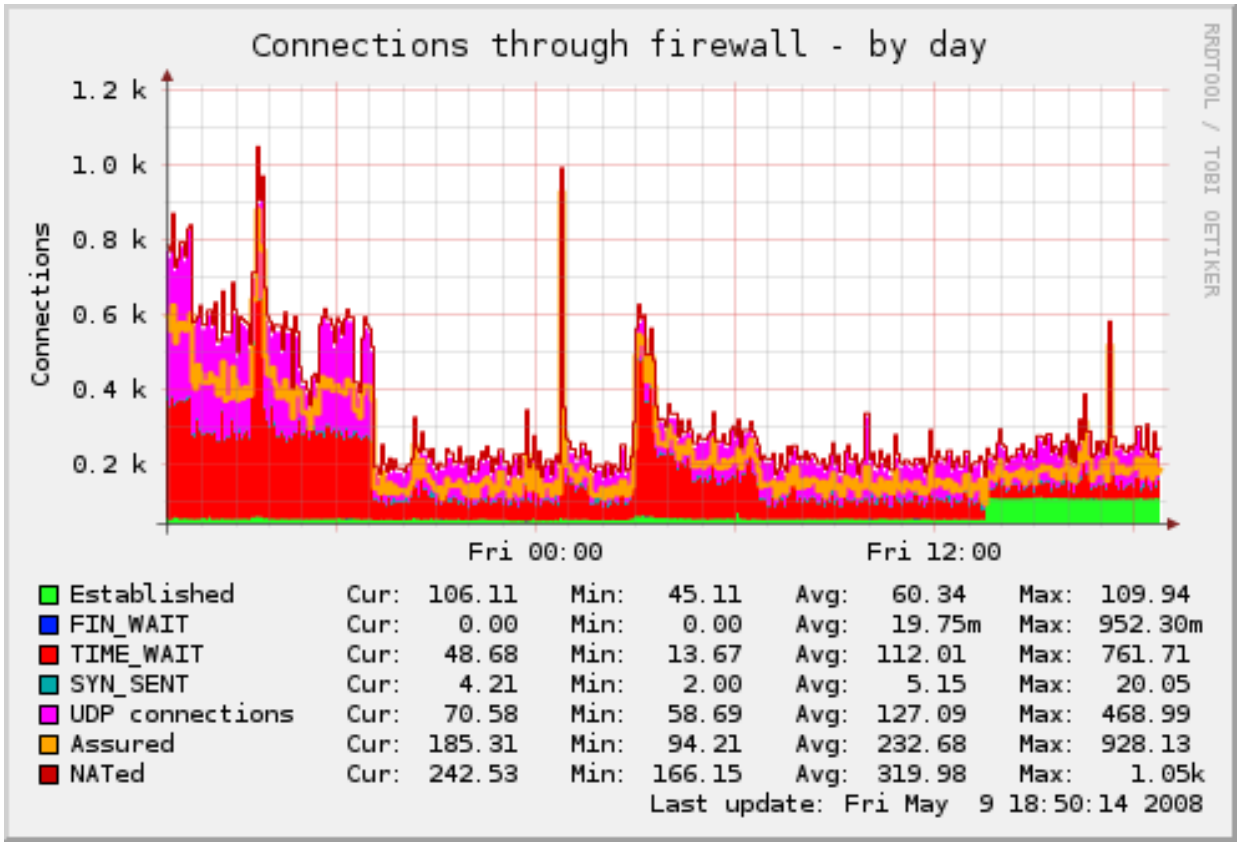
Leyendo gráficas como mecanismo de aprendizaje

- Leer las gráficas generadas por Munin nos puede llevar a aprender y entender muchos aspectos del funcionamiento de nuestro sistema
- Es importante que revisemos todos los aspectos que llamen nuestra atención — Pueden llevarnos a sorprendentes revelaciones

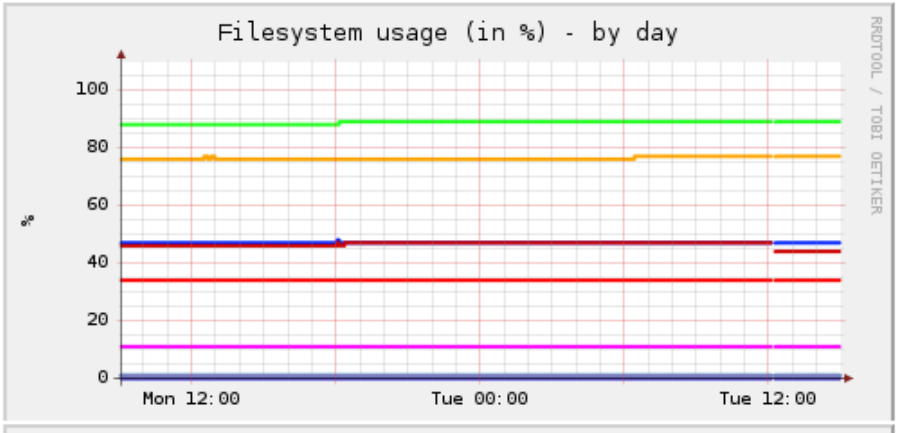
Uso de CPU

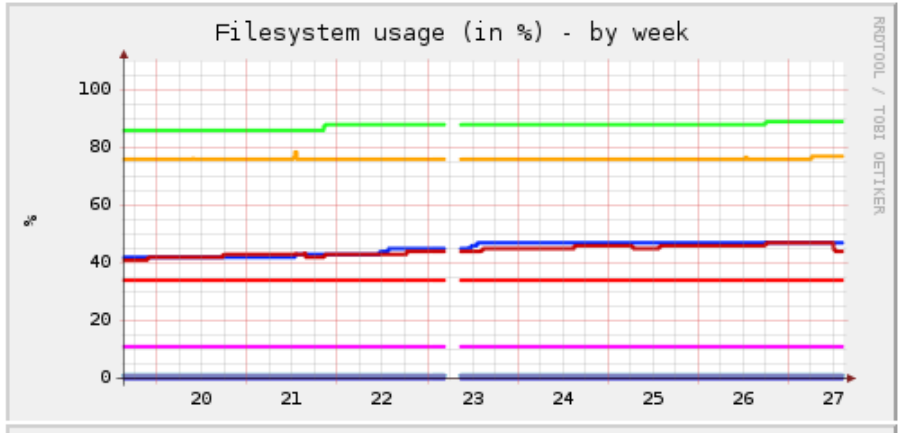

CPU usage - by day
RRDTOOL / TOBI OETIKER
%
100
80
60
40
20
0
Thu 12:00
Fri 00:00
Fri 12:00
system Cur: 0.94 Min: 0.90 Avg: 3.04 Max: 20.18
user Cur: 0.99 Min: 0.69 Avg: 3.21 Max: 22.51
nice Cur: 1.55 Min: 0.27 Avg: 1.96 Max: 6.99
idle Cur: 96.31 Min: 0.01 Avg: 85.16 Max: 96.86
iowait Cur: 0.18 Min: 0.00 Avg: 6.42 Max: 90.23
irq Cur: 0.01 Min: 0.00 Avg: 0.14 Max: 1.83
softirq Cur: 0.00 Min: 0.00 Avg: 0.08 Max: 0.71
Last update: Fri May 9 12:50:04 2008

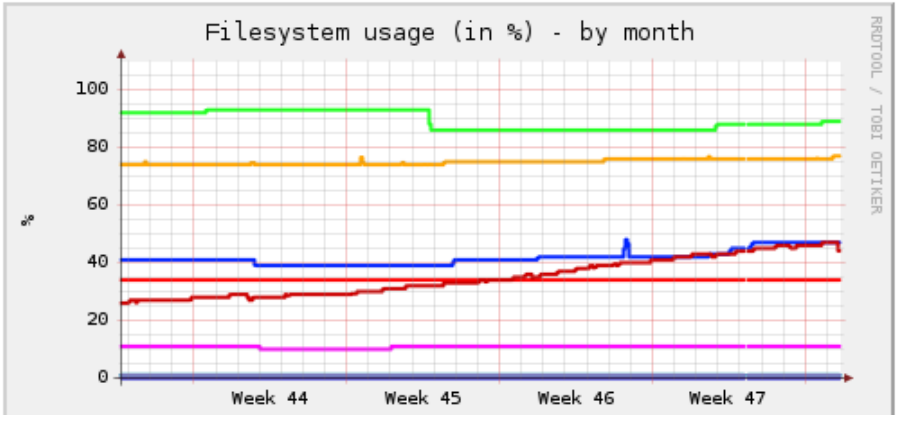
Número y estado de conexiones manejadas por el firewall

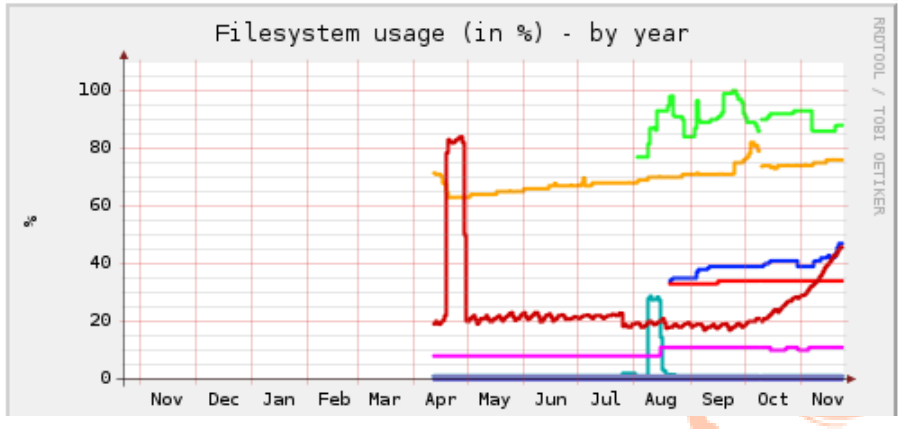


Uso (anual) del sistema de archivos

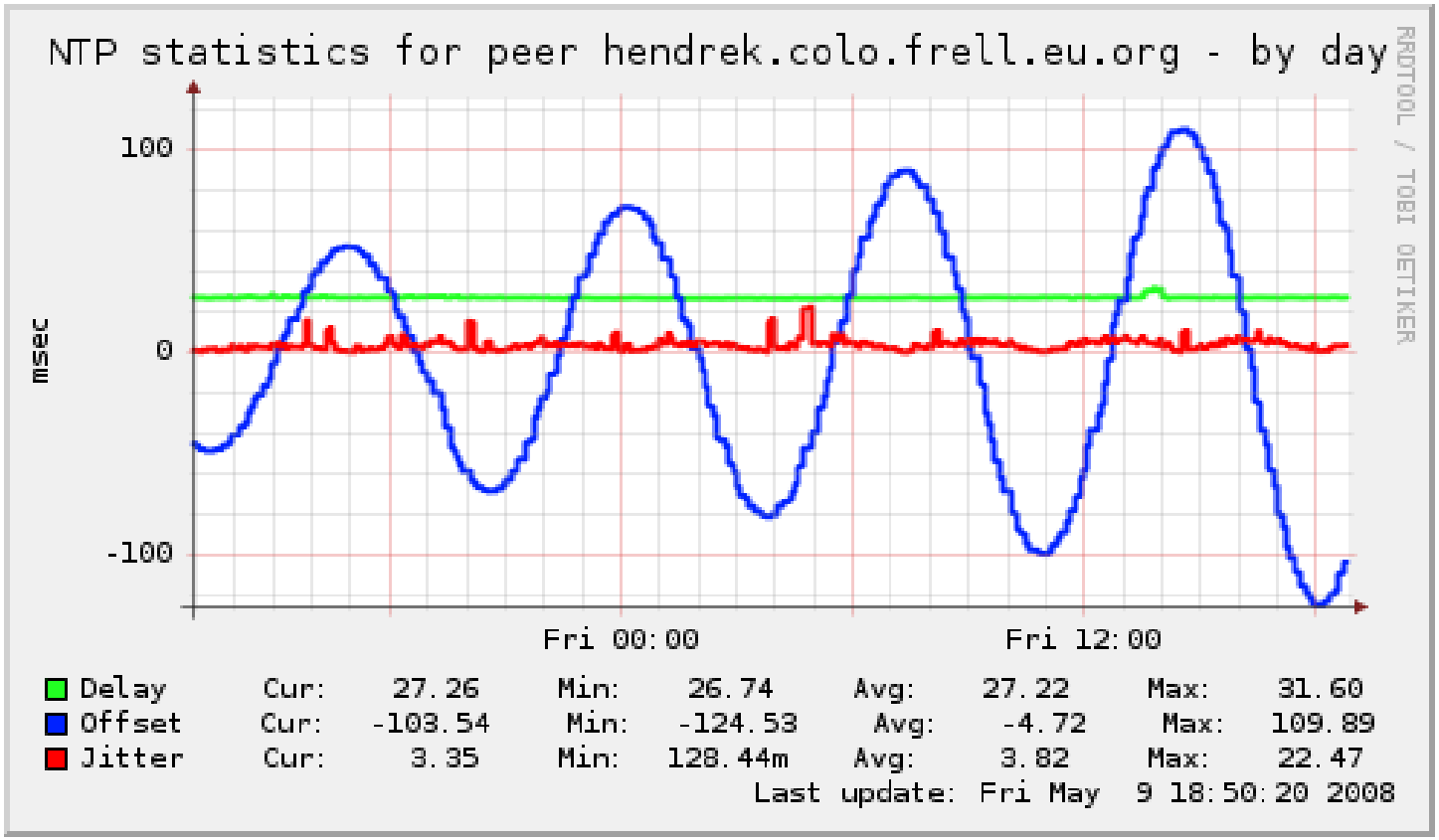




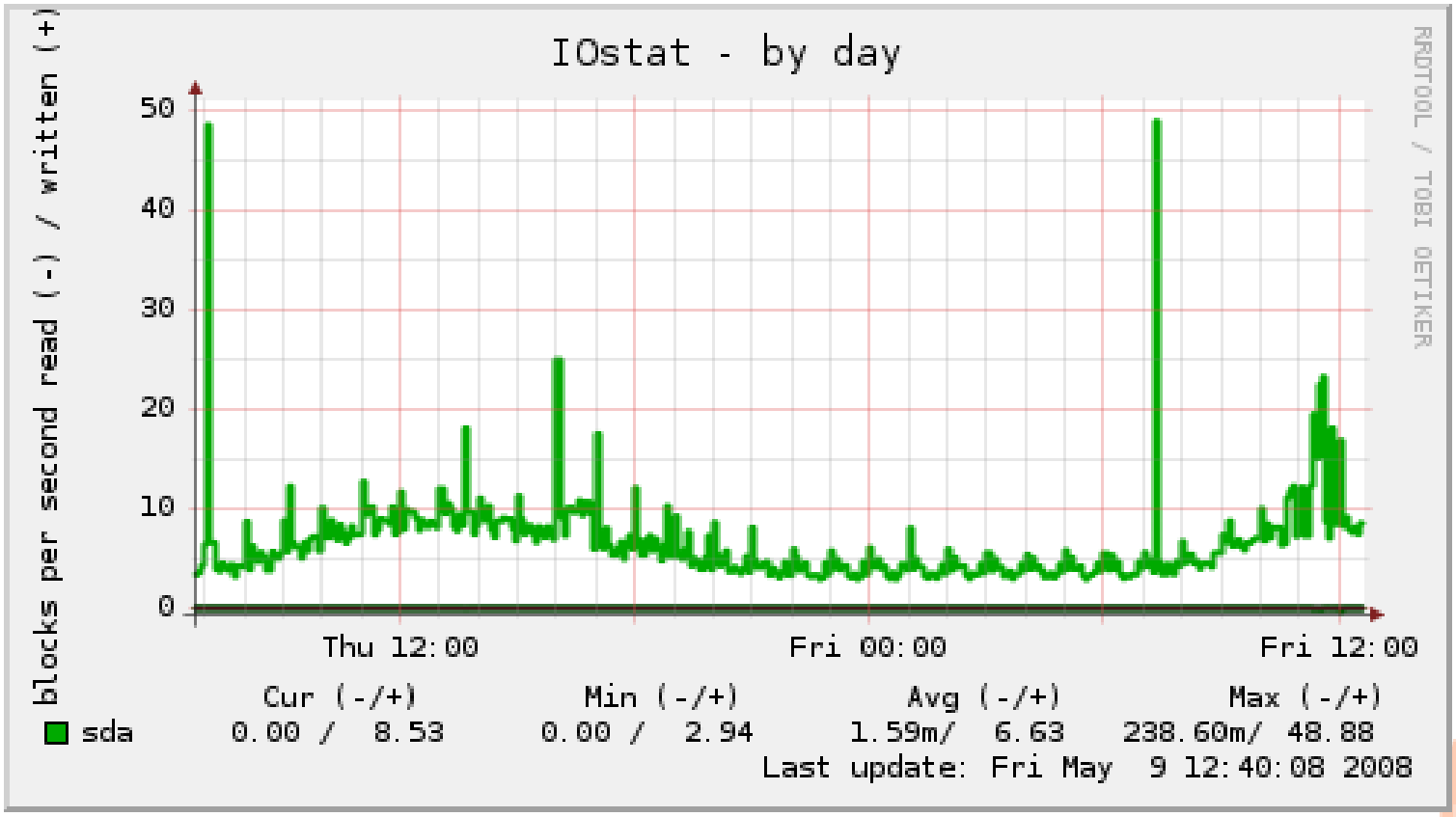




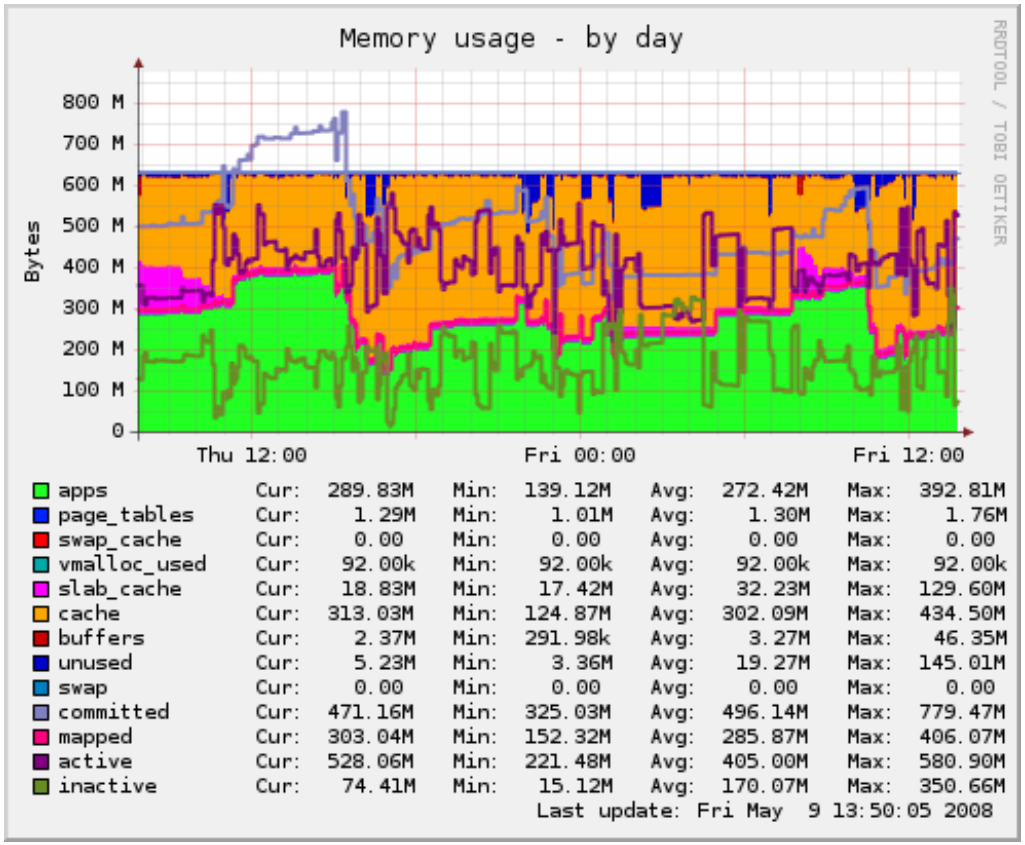
Sincronización con el reloj atómico (NTP)



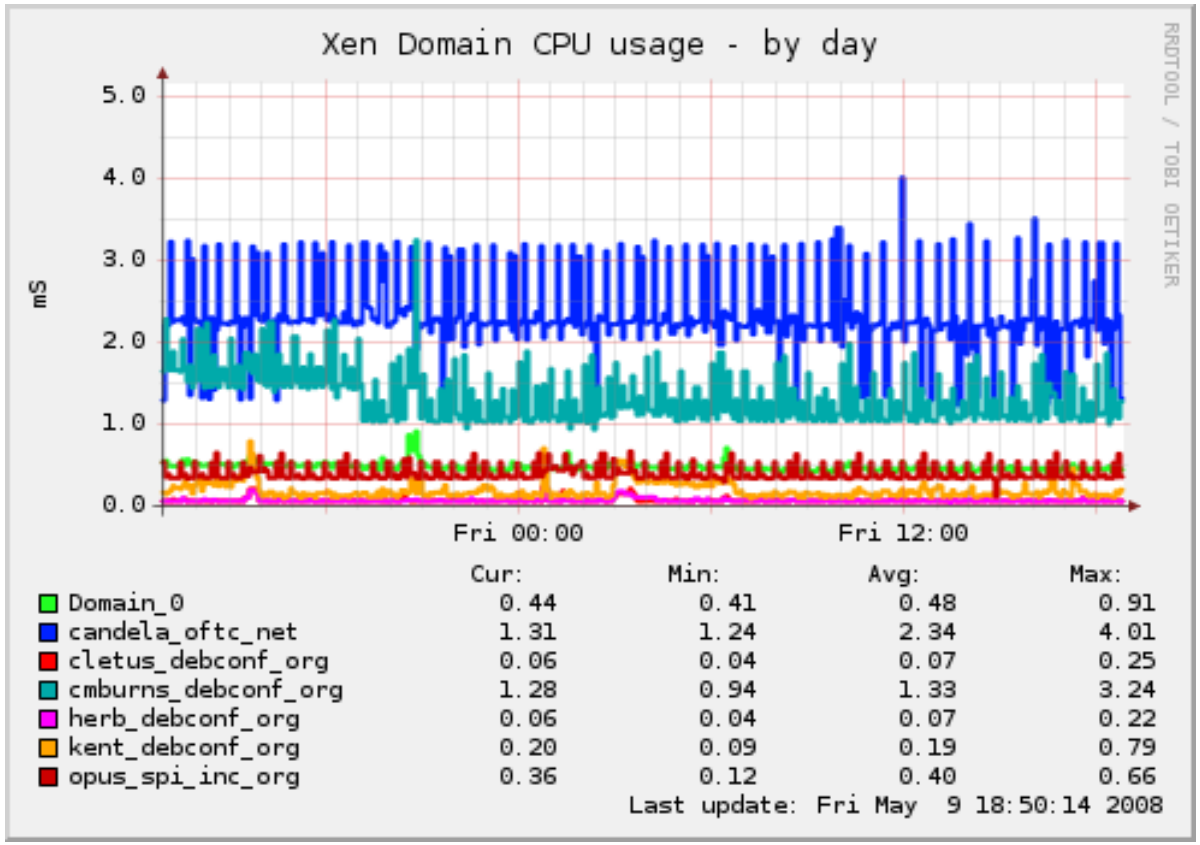
Uso del bus de entrada/salida



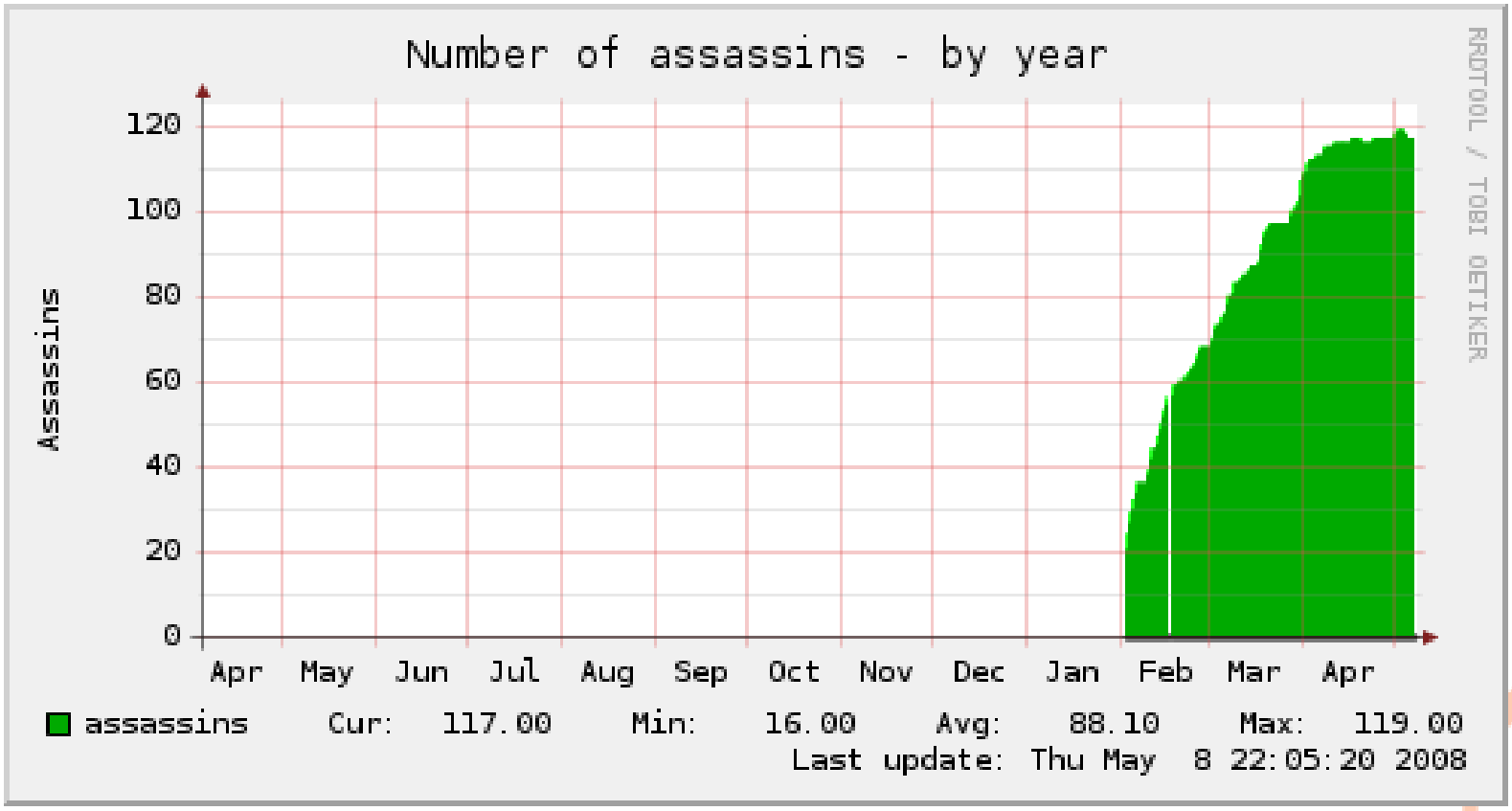
Uso de memoria



Uso de CPU por DomU en Xen



Número de *asesinos* registrados a lo largo del año



Contents

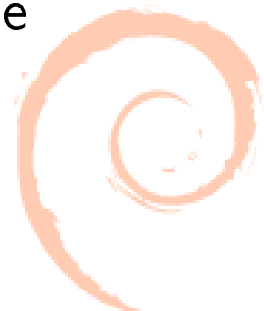
¡El cielo es el límite!

- Probablemente la mayor fuerza de Munin es lo fácil que resulta implementar y desplegar nuestros propios plugins
- Los plugins pueden implementarse en cualquier lenguaje, basta con ofrecer una interfaz muy simple
- Cualquier cosa cuantificable es monitoreable
- Para realmente aprovecharlo en realidad, nos conviene aprender las sutilezas de RRDtool

Lógica de operación de un plugin

- Cada plugin es ejecutado, como un script cualquiera.
- El plugin corre con los privilegios de usuario definidos en su entrada en /etc/munin/plugin-conf.d/
- Su modo de operación lo determina el único parámetro

`autoconf`	¿Es capaz de autoconfigurarse en base al entorno? (`yes` / `no` y estado de salida)
`suggest`	¿En qué casos se *autosugerirá* al cliente Munin? (por estado de salida)
`config`	¿Qué parámetros graficará? (Descripción de la gráfica, etiquetas de los ejes y las variables, e indicaciones de formato a RRDtool
Sin parámetros	Entrega los resultados del monitoreo en cuestión



¿Y cómo obtengo los datos?

... Como quieras.

- ¿Datos de salud/parámetros de operación del núcleo? `cat` + `grep` sobre `/proc` y similares
- ¿Estado de los recursos del sistema? Tomamos la salida de la ejecución de diferentes comandos `df`, `du`, etc.
- ¿Datos obtenidos de la red (p.ej. monitoreo a sitios remotos)? A través de un script en nuestro lenguaje favorito
- ¿Niveles de bases de datos? Scripts que interroguen a la BD en cuestión, en nuestro lenguaje favorito
- ... ¡Lo que sea que nos pueda generar salida cuantificable!

¿Y cómo obtengo los datos?

... Como quieras.

- ¿Datos de salud/parámetros de operación del núcleo? `cat` + `grep` sobre `/proc` y similares
- ¿Estado de los recursos del sistema? Tomamos la salida de la ejecución de diferentes comandos `df`, `du`, etc.
- ¿Datos obtenidos de la red (p.ej. monitoreo a sitios remotos)? A través de un script en nuestro lenguaje favorito
- ¿Niveles de bases de datos? Scripts que interroguen a la BD en cuestión, en nuestro lenguaje favorito
- ... ¡Lo que sea que nos pueda generar salida cuantificable!

Cosas a tomar en cuenta con nuestros plugins

- ¿Con qué privilegios va a correr? ¿Puedo *obligar* a que el script no sea llamado como root?
- ¿Qué tiempo de ejecución tiene cada plugin? Recuerden que voy a tener una invocación de *todos* mis plugins cada cinco minutos. ¿Puedo separarlo en hilos o procesos?
- ¿Qué tan sensibles son estos datos? Recuerden que Munin **no implementa autenticación ni cifado** — Pueden configurarlo p.ej. sobre interfaces ligadas a una VPN.

Contents

Concluyendo

Munin es...

- Bonito
- Divertido
- Una herramienta para monitorear nuestros sistemas
- Un gran recurso para *aprender* acerca de nuestros sistemas
- Una maravillosa herramienta para apantallar al jefe
- Un pedazo de mitología nórdica :-}
- Una *excelente* manera de perder el tiempo

¿Dudas?

¡Gracias!

Gunnar Wolf — gwolf@gwolf.org

http://www.gwolf.org/soft/munin