

Ingeniería de Servidores (2014-2015)
GRADO EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

Memoria Práctica 3

Jesus Checa Hidalgo

25 de noviembre de 2014

Índice

1. Cuestión 1	3
1.1. ¿Qué archivo le permite ver qué programas se han instalado con el gestor de paquetes?	3
1.2. ¿Qué significan las terminaciones .1.gz o .2.gz de los archivos en ese directorio?	3
2. Cuestión 2	3
2.1. ¿Qué archivo ha de modificar para programar una tarea? Escriba la línea necesaria para ejecutar una vez al día una copia del directorio /codigo a /seguridad/fecha donde fecha es la fecha actual (puede usar el comando date).	3
3. Cuestión 3	4
3.1. Pruebe a ejecutar el comando, conectar un dispositivo USB y vuelva a ejecutar el comando. Copie y pegue la salida del comando. (considere usar dmesg tail). Comente qué observa en la información mostrada.	4
4. Question 4	5
4.1. Ejecute el monitor de “System Performance” y muestre el resultado. Incluya capturas de pantalla comentando la información que aparece.	5
5. Question 5	8
5.1. Cree un recopilador de datos definido por el usuario (modo avanzado) que incluya tanto el contador de rendimiento como los datos de seguimiento: Todos los referentes al procesador, al proceso y al servicio web. Intervalo de muestra 15 segundos Almacene el resultado en el directorio Escritorio/logs Incluya las capturas de pantalla de cada paso.	8
6. Question 6	13
6.1. Instale alguno de los monitores comentados arriba en su máquina y pruebe a ejecutarlos (tenga en cuenta que si lo hace en la máquina virtual, los resultados pueden no ser realistas). Alternativamente, busque otros monitores para hardware comerciales o de código abierto para Windows y Linux.	13
7. Cuestión 7	16
7.1. Visite la web del proyecto y acceda a la demo que proporcionan (http://demo.munin-monitoring.org/) donde se muestra cómo monitorizan un servidor. Monitorice varios parámetros y haga capturas de pantalla de lo que está mostrando comentando qué observa.	16
8. Cuestión 8	17
8.1. Escriba un breve resumen sobre alguno de los artículos donde se muestra el uso de strace o busque otro y coméntelo.	17

9. Question 9	18
9.1. Acceda a la consola mysql (o a través de phpMyAdmin) y muestre el resultado de mostrar el "profile" de una consulta (la creación de la BD y la consulta la puede hacer libremente).	18
10. Question opcional 1	21
10.1. Indique qué comandos ha utilizado para realizarlo así como capturas de pantalla del proceso de reconstrucción del RAID.	21
11. Question opcional 2	25
11.1. Instale Nagios en su sistema (el que prefiera) documentando el proceso y muestre el resultado de la monitorización de su sistema comentando qué aparece. . .	25
12. Cuestión opcional 3	29
12.1. Haga lo mismo que con Munin	29

Índice de figuras

3.1. Salida de <code>dmesg</code> antes de enchufar el dispositivo	4
3.2. Despues de conectar el USB	4
4.1. Pantalla de bienvenida de <code>perfmon(I)</code>	5
4.2. Pantalla de bienvenida de <code>perfmon(II)</code>	6
4.3. Gráfico de carga del sistema en <code>perfmon</code>	6
4.4. Recopilador de datos del sistema(I)	7
4.5. Recopilador de datos del sistema(II)	7
5.1. Elegimos un nombre para el recopilador y seleccionamos crear manualmente . .	8
5.2. Seleccionamos los registros que queremos contemplar	8
5.3. Elegimos "Agregar..." para añadir parametros	9
5.4. Para cada uno de los parametros, lo seleccionamos, seleccionamos "_Total", y "Agregar"	9
5.5. Antes de continuar, nos aseguramos que el intervalo es el deseado	10
5.6. No queremos seleccionar ningun proveedor de seguimiento, solo seleccionamos siguiente	10
5.7. Seleccionamos la ruta deseada para guardar los datos	11
5.8. Aqui podemos elegir que usuario ejecuta el log. Seleccionamos "Iniciar ahora" y damos a Finalizar	11
5.9. Informe definido por usuario	12
6.1. <code>lm_sensors</code> , muestra de temperaturas de CPU	13
6.2. Ejecucion de <code>hddtemp</code>	13
6.3. Seccion de CPU de CPU-G	14
6.4. Informacion de la placa base de CPU-G	14
6.5. Informacion de RAM de CPU-G	15
6.6. Informacion adicional del sistema en CPU-G	15
7.1. Entrada/salida por dispositivo	16

7.2. Accesos a apache	16
7.3. Transferencias por la interfaz eth0	17
7.4. Threads activos	17
9.1. Consola de comandos de PHPMyAdmin	18
9.2. SHOW profiles, que nos muestra todas las consultas registradas.	19
9.3. SHOW profile, que muestra el profiling de la consulta mas reciente.	19
9.4. SHOW profile, con “for query n” nos muestra el profiling relativo a esa consulta n	20
9.5. SHOW profile con un especificador.	20
9.6. SHOW profile con el especificador ALL	20
10.1. /proc/mdstat antes de fallar ningun disco	21
10.2. RAID con disco antiguo	21
10.3. RAID con el nuevo disco	22
10.4. Propmt de initramfs	22
10.5. /proc/mdstat con un disco dañado o cambiado	22
10.6. mdadm con el modificador -detail	23
10.7. Proceso de reconstruccion consultado con mdstat	24
10.8. Proceso de reconstruccion consultado con watch y /proc/mdstat	24
11.1. Resumen de la configuracion de Nagios	26
11.2. Salida de make install	26
11.3. Comprobación de la configuración de Nagios	27
11.4. Pantalla inicial de Nagios	28
11.5. Resumen del sistema	29
12.1. Carga de trabajo del grid de Wikimedia completo	30
12.2. Carga de trabajo de un cluster del grid	30
12.3. Carga de trabajo de un nodo de un cluster	31

1. Cuestión 1

1.1. ¿Qué archivo le permite ver qué programas se han instalado con el gestor de paquetes?

Para los paquetes instalados con yum, podemos consultarlo en el fichero `/var/log/yum.log`¹

1.2. ¿Qué significan las terminaciones `.1.gz` o `.2.gz` de los archivos en ese directorio?

Estos archivos son logs antiguos. Como los logs van creciendo mas y mas, el sistema les aplica una rotacion, poniendole la extension `.1.gz`, `.2.gz`, etc, y crea uno nuevo, vacío. De esta forma evitamos que los logs desborden el almacenamiento. Cuanto mayor sea el numero, mas antiguo es el log².

2. Cuestión 2

2.1. ¿Qué archivo ha de modificar para programar una tarea? Escriba la línea necesaria para ejecutar una vez al día una copia del directorio `/codigo` a `/seguridad/fecha` donde `fecha` es la fecha actual (puede usar el comando `date`).

Primero tenemos que crear el script que efectuará la tarea:

```
1 #!/bin/bash
2
3 fecha=$(date +%m_%d_%y)
4 mkdir -p "seguridad/${fecha}"
5 cp -r ~/codigo/* "~/seguridad/${fecha}"
```

Para que se ejecute una vez al dia, usaremos el demonio cron³, y mas concretamente con el archivo `/etc/crontab`, en donde se guardan las tareas periódicas:

```
1 # echo "0 15 * * * user sh ~/.scripts/backupcodigo.sh" >> /etc/crontab
```

Con esta linea, el sistema ejecutará a las 15:00 de cualquier dia de cualquier mes, sea el dia de la semana que sea, la orden `sh ~/.scripts/backupcodigo.sh` como el usuario `user`. Obviamente, el script esta guardado en el directorio `“.scripts”` de la carpeta personal del usuario.

¹<http://linuxnotes.us/archives/510>

²<http://llozadac.wordpress.com/administracion-web/logs-de-linux/>

³Informacion extraida de las prácticas de la asignatura Sistemas Operativos, Modulo I: Administracion de Linux - Sesión 4: Automatización de tareas

3. Cuestión 3

3.1. Pruebe a ejecutar el comando, conectar un dispositivo USB y vuelva a ejecutar el comando. Copie y pegue la salida del comando. (considere usar `dmesg | tail`). Comente qué observa en la información mostrada.

Primero ejecuto `dmesg` sin insertar el USB(figura 3.1), y veo la salida. Conecto el USB (una interfaz de audio), vuelvo a ejecutar `dmesg` y me muestra varios datos(figura 3.2, entre ellos a qué puerto usb se ha conectado (en mi caso al segundo USB3.0), que dispositivo estamos conectando (USB AUDIO CODEC), el código identificador del producto (Product ID y Vendor ID), y el driver empleado (`snd-usb-audio` en mi caso).

```
[ 8270.564821] IPv6: ADDRCONF(NETDEV_UP): enp8s0: link is not ready
[ 8270.576558] IPv6: ADDRCONF(NETDEV_UP): wlp10s0: link is not ready
[ 8274.166075] wlp10s0: authenticate with [REDACTED]
[ 8274.171674] wlp10s0: send auth to [REDACTED] (try 1/3)
[ 8274.173604] wlp10s0: authenticated
[ 8274.175442] wlp10s0: associate with [REDACTED] (try 1/3)
[ 8274.178918] wlp10s0: RX AssocResp from [REDACTED] (capab=0x411 status=0 aid=5)
[ 8274.178960] IPv6: ADDRCONF(NETDEV_CHANGE): wlp10s0: link becomes ready
[ 8274.178981] wlp10s0: associated
[ 8407.465284] mce: [Hardware Error]: Machine check events logged
[jesus@alien ~]$
```

Figura 3.1: Salida de `dmesg` antes de enchufar el dispositivo

```
[ 8270.564821] IPv6: ADDRCONF(NETDEV_UP): enp8s0: link is not ready
[ 8270.576558] IPv6: ADDRCONF(NETDEV_UP): wlp10s0: link is not ready
[ 8274.166075] wlp10s0: authenticate with [REDACTED]
[ 8274.171674] wlp10s0: send auth to [REDACTED] (try 1/3)
[ 8274.173604] wlp10s0: authenticated
[ 8274.175442] wlp10s0: associate with [REDACTED] (try 1/3)
[ 8274.178918] wlp10s0: RX AssocResp from [REDACTED] (capab=0x411 status=0 aid=5)
[ 8274.178960] IPv6: ADDRCONF(NETDEV_CHANGE): wlp10s0: link becomes ready
[ 8274.178981] wlp10s0: associated
[ 8407.465284] mce: [Hardware Error]: Machine check events logged
[ 9253.134070] usb 3-2: new full-speed USB device number 3 using xhci_hcd
[ 9253.310728] usb 3-2: ep 0x85 - rounding interval to 64 microframes, ep desc says 80 microframes
[ 9253.311484] input: Burr-Brown from TI USB Audio CODEC as /devices/pci0000:00/0000:00:14.0/usb3/3-2/3-2:1.3/0003:08BB:2902.0006/input/input17
[ 9253.311642] hid-generic 0003:08BB:2902.0006: input,hidraw3: USB HID v1.00 Device [Burr-Brown from TI USB Audio CODEC] on usb-0000:00:14.0-2/input3
[ 9253.860943] usbcore: registered new interface driver snd-usb-audio
[jesus@alien ~]$
```

Figura 3.2: Después de conectar el USB

4. Cuestion 4

4.1. Ejecute el monitor de “System Performance” y muestre el resultado. Incluya capturas de pantalla comentando la información que aparece.

En la primera pantalla nos aparece un resumen del sistema e información sobre dónde podemos encontrar ayuda (Figuras 4.1 y 4.2). También tenemos un monitor de rendimiento que mide la carga del sistema por segundo (Figura 4.3)

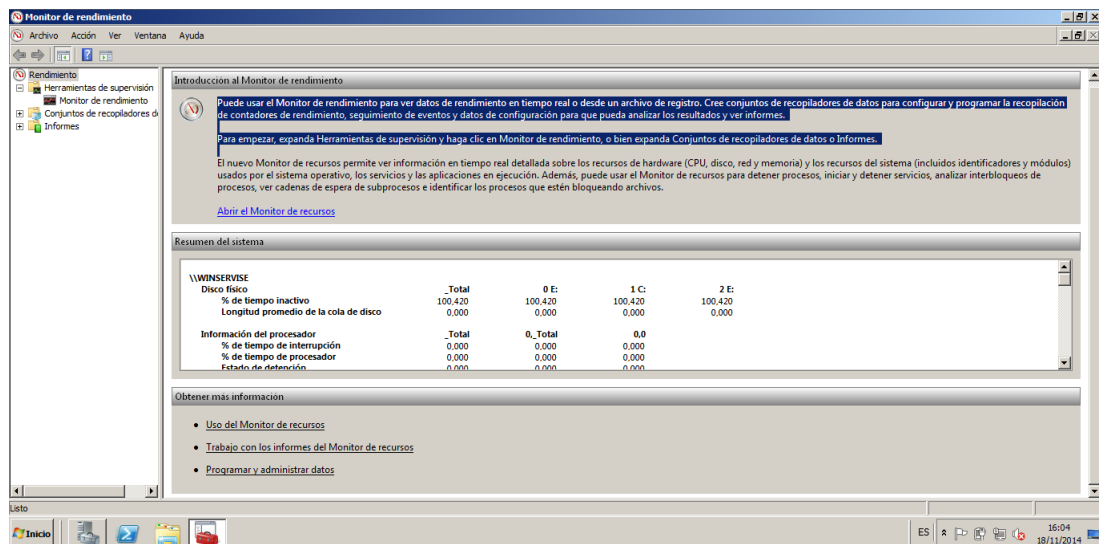


Figura 4.1: Pantalla de bienvenida de perfmon(I)

Una vez iniciamos el recopilador de datos del sistema vemos un resumen del sistema mas completo, resultados de diagnóstico de rendimiento (porcentaje de carga de recursos)(Figura 4.4) y luego un informe pormenorizado para cada uno de los recursos (CPU, red, memoria, y discos), con detalles varios (Figura 4.5)

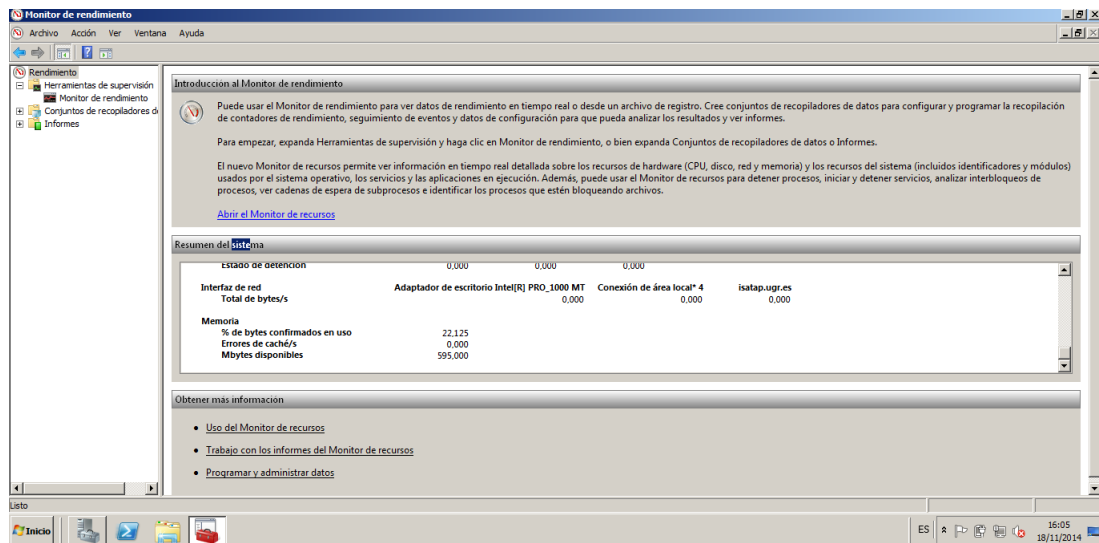


Figura 4.2: Pantalla de bienvenida de perfmon(II)

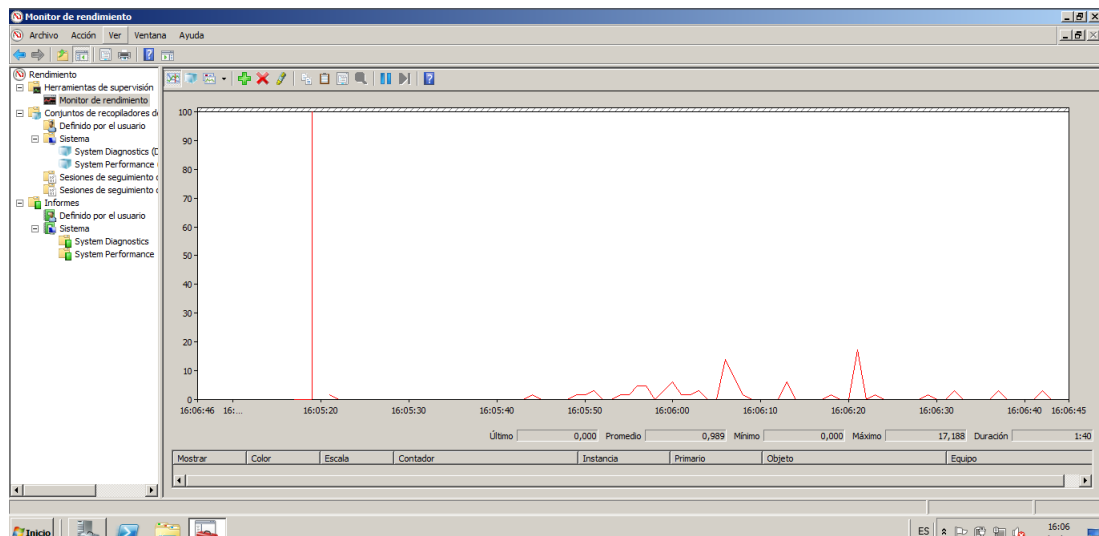


Figura 4.3: Gráfico de carga del sistema en perfmon

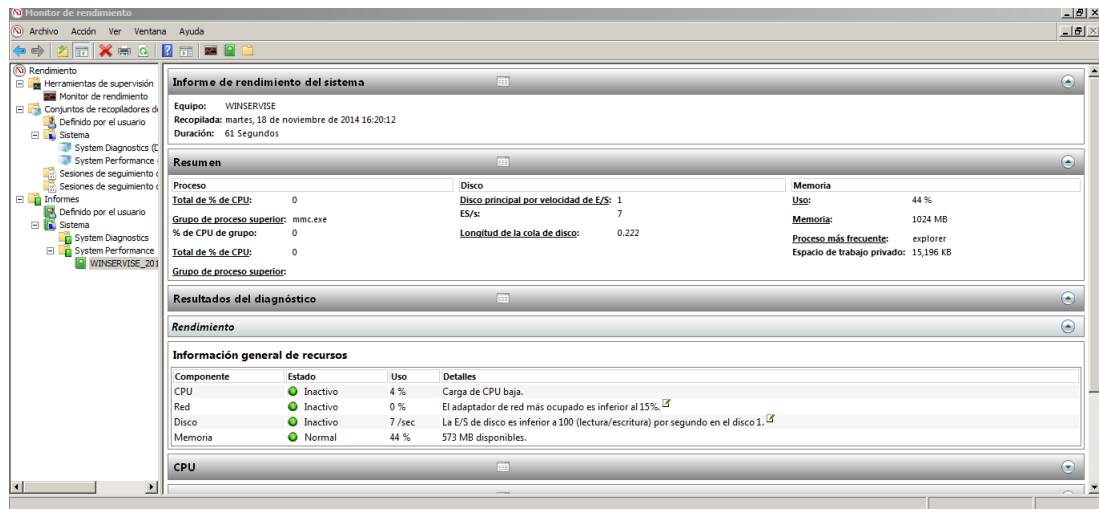


Figura 4.4: Recopilador de datos del sistema(I)

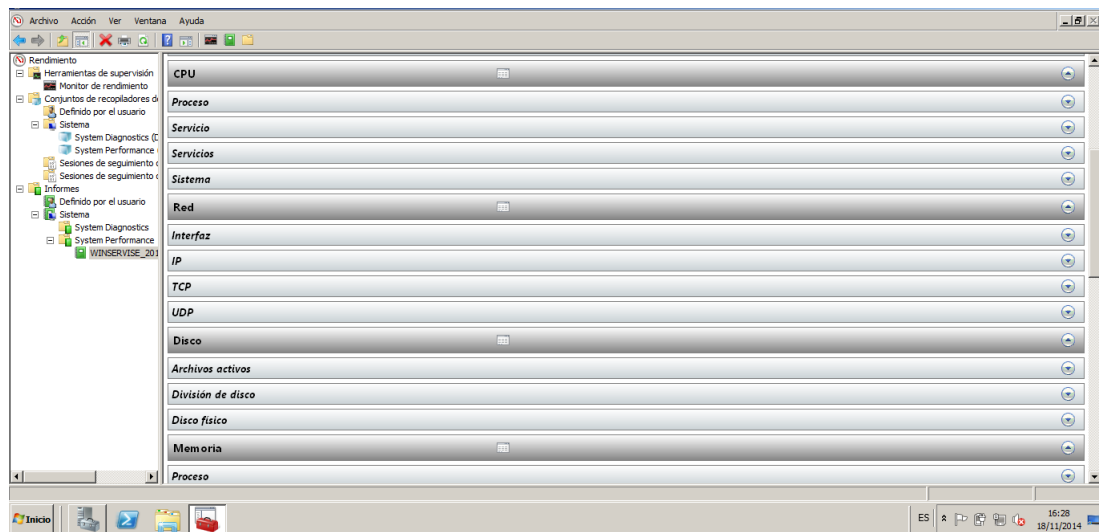


Figura 4.5: Recopilador de datos del sistema(II)

5. Cuestion 5

- 5.1. Cree un recopilador de datos definido por el usuario (modo avanzado) que incluya tanto el contador de rendimiento como los datos de seguimiento: Todos los referentes al procesador, al proceso y al servicio web. Intervalo de muestra 15 segundos. Almacene el resultado en el directorio Escritorio/logs. Incluya las capturas de pantalla de cada paso.

En la barra de la izquierda, en Conjuntos de recopiladores de datos, hacemos clic derecho sobre "Definido por el usuario" seleccionamos "Nuevo", Conjuntos de recopiladores de datos". Las capturas de la 5.1 a la 5.8 reflejan la creación del recopilador.

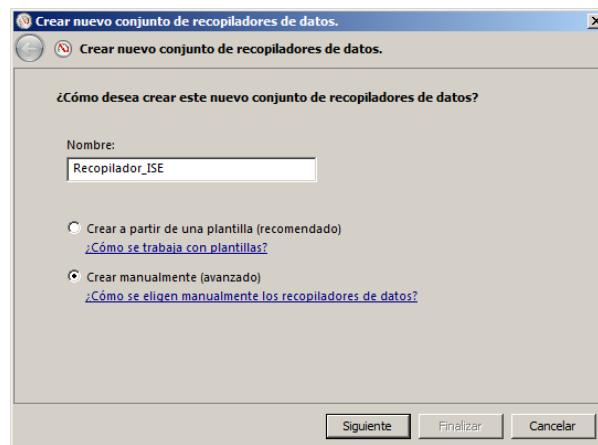


Figura 5.1: Elegimos un nombre para el recopilador y seleccionamos crear manualmente

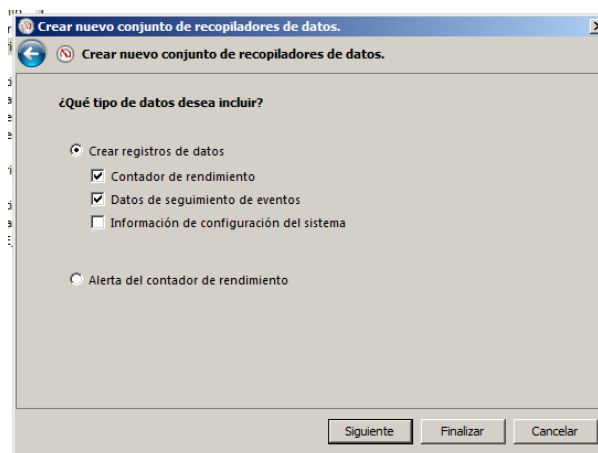


Figura 5.2: Seleccionamos los registros que queremos contemplar

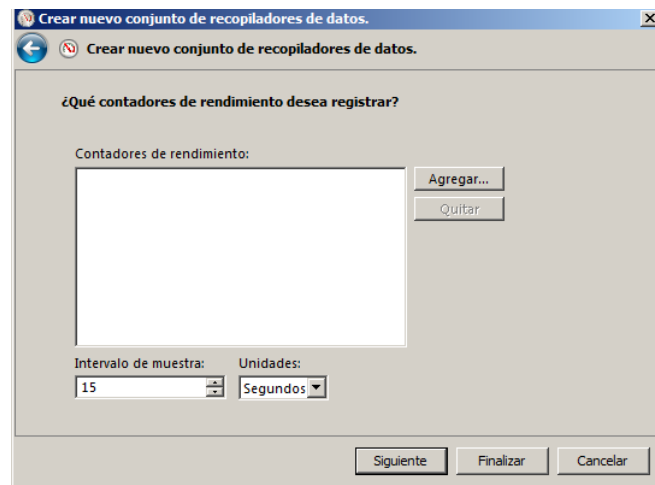


Figura 5.3: Elegimos “Agregar...” para añadir parámetros

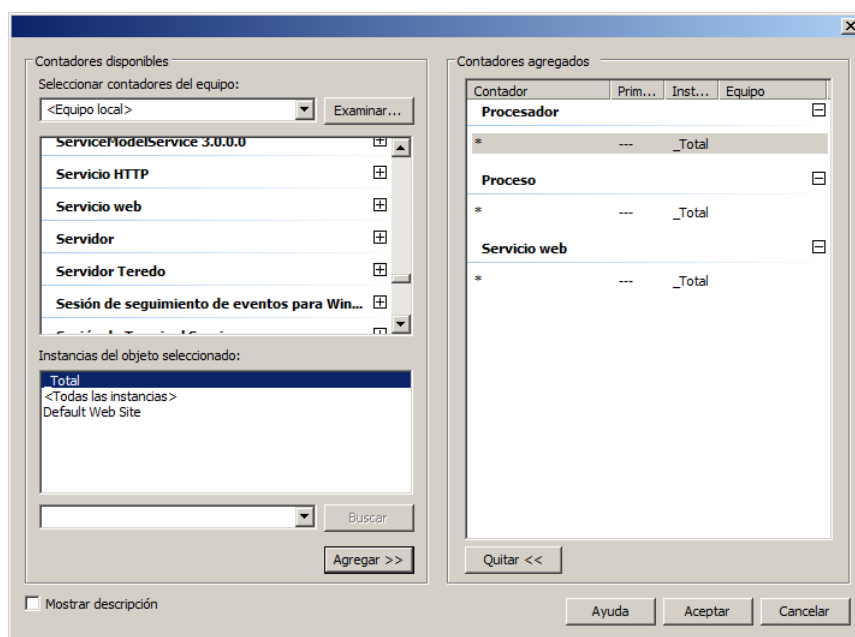


Figura 5.4: Para cada uno de los parámetros, lo seleccionamos, seleccionamos “_Total”, y “Agregar”

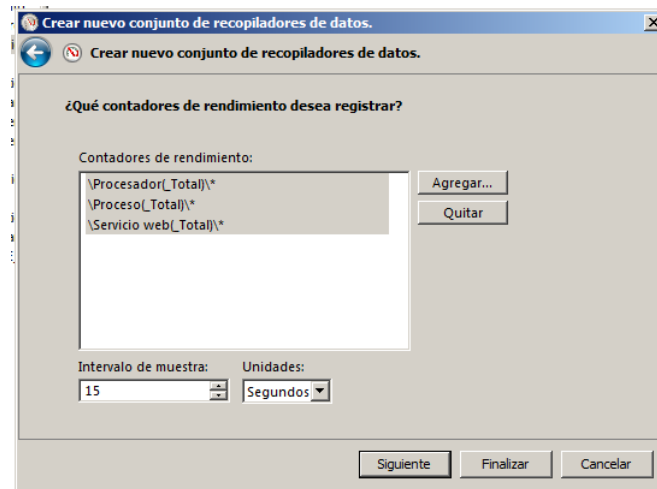


Figura 5.5: Antes de continuar, nos aseguramos que el intervalo es el deseado

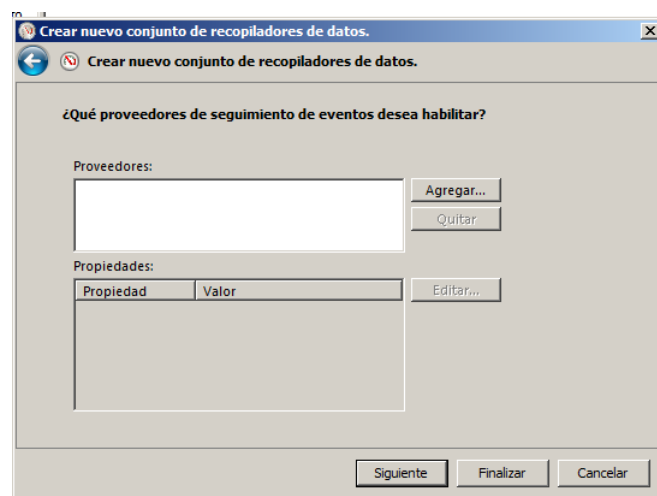


Figura 5.6: No queremos seleccionar ningún proveedor de seguimiento, solo seleccionamos siguiente

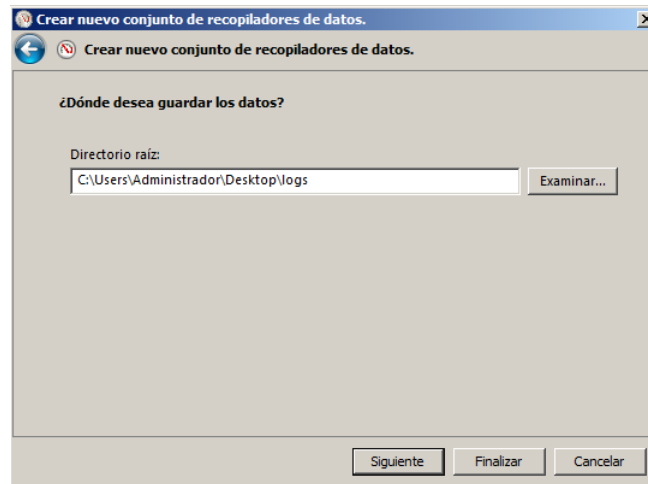


Figura 5.7: Seleccionamos la ruta deseada para guardar los datos

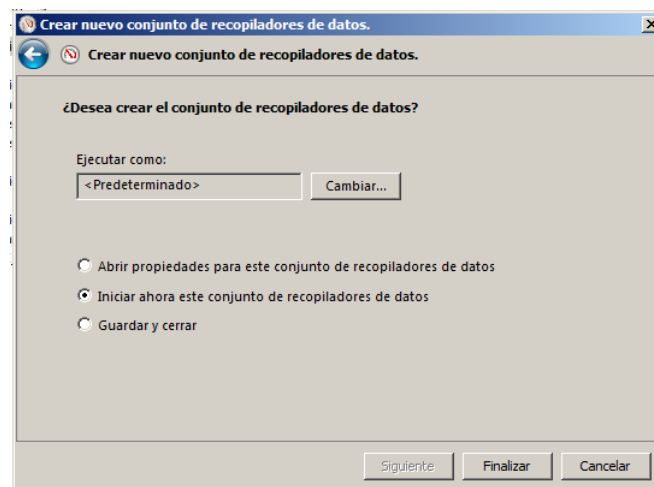


Figura 5.8: Aquí podemos elegir que usuario ejecuta el log. Seleccionamos “Iniciar ahora” y damos a Finalizar

Despues de creado, podemos ver el informe en “Informes ->Definido por el usuario ->NombredelRecopilador” (Figura 5.9)

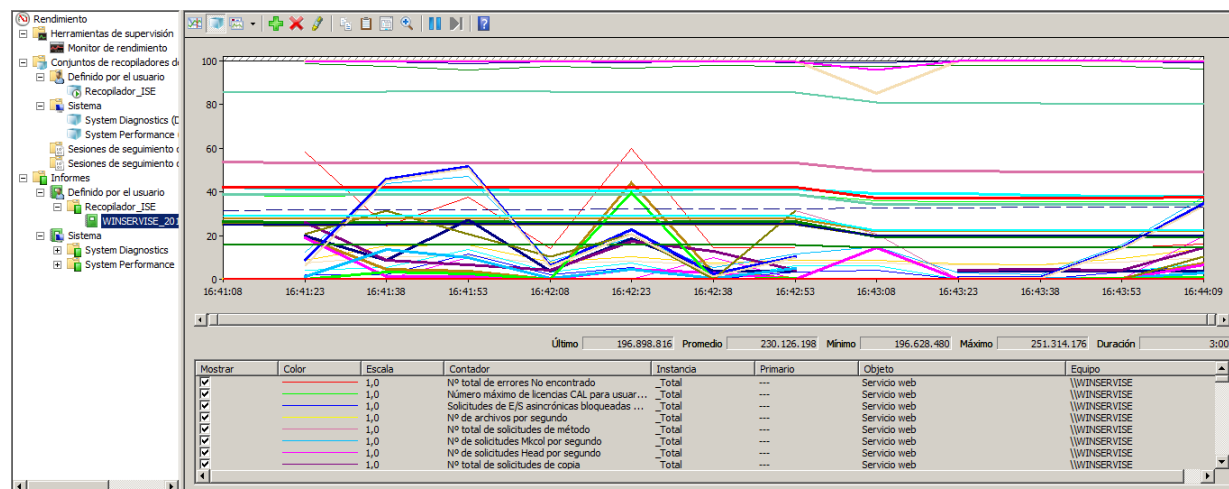


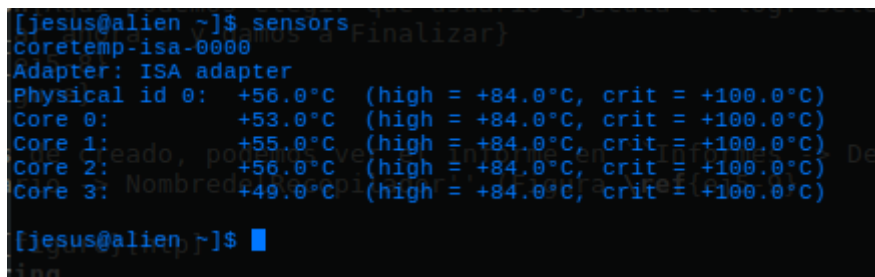
Figura 5.9: Informe definido por usuario

6. Cuestion 6

- 6.1. Instale alguno de los monitores comentados arriba en su máquina y pruebe a ejecutarlos (tenga en cuenta que si lo hace en la máquina virtual, los resultados pueden no ser realistas). Alternativamente, busque otros monitores para hardware comerciales o de código abierto para Windows y Linux.**

Hddtemp y lm-sensors son herramientas que tengo en mi equipo y que uso regularmente para controlar las temperaturas (también para incrustar datos en scripts de conky).

Datos de lm_sensors:

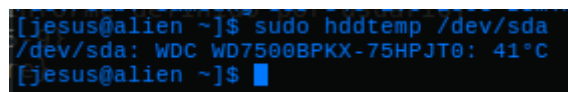


```
[jesus@alien ~]$ sensors
coretemp-isa-0000
Adapter: ISA adapter
Physical id 0: +56.0°C (high = +84.0°C, crit = +100.0°C)
Core 0: +53.0°C (high = +84.0°C, crit = +100.0°C)
Core 1: +55.0°C (high = +84.0°C, crit = +100.0°C)
Core 2: +56.0°C (high = +84.0°C, crit = +100.0°C)
Core 3: +49.0°C (high = +84.0°C, crit = +100.0°C)

[jesus@alien ~]$
```

Figura 6.1: lm_sensors, muestra de temperaturas de CPU

Datos mostrados por hddtemp:



```
[jesus@alien ~]$ sudo hddtemp /dev/sda
/dev/sda: WDC WD7500BPKX-75HPJT0: 41°C
[jesus@alien ~]$
```

Figura 6.2: Ejecucion de hddtemp

Otro software de monitorización que he usado en algunas ocasiones es CPU-G, una versión con GTK de CPU-Z. Para instalarlo en ArchLinux podemos hacerlo directamente desde el AUR con yaourt:

```
1 $ yaourt -S cpu-g
```

Nos ofrece información sobre los distintos componentes del equipo. Para CPU nos muestra tanto marca, modelo, y frecuencia, como capacidades de caché y otros datos (Figura 6.3).

Tenemos también información sobre el modelo de placa base, y versión y fabricante de la BIOS (Figura 6.4).

En cuanto a memoria RAM, nos muestra la memoria total, la usada, la activa... etc (figura 6.5). También tendremos información variada sobre el sistema operativo, como la arquitectura, kernel, versión de GCC, de Xorg, o tiempo de funcionamiento (figura 6.6).

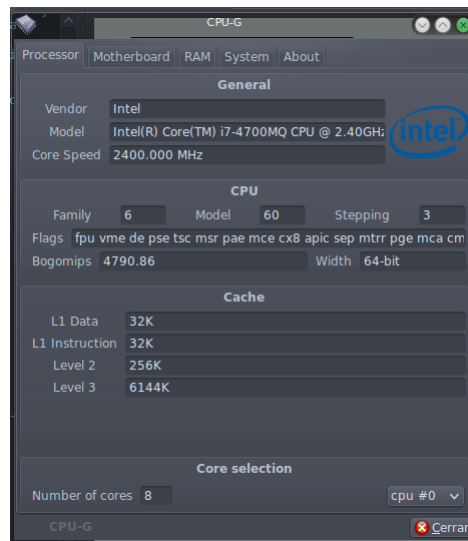


Figura 6.3: Seccion de CPU de CPU-G

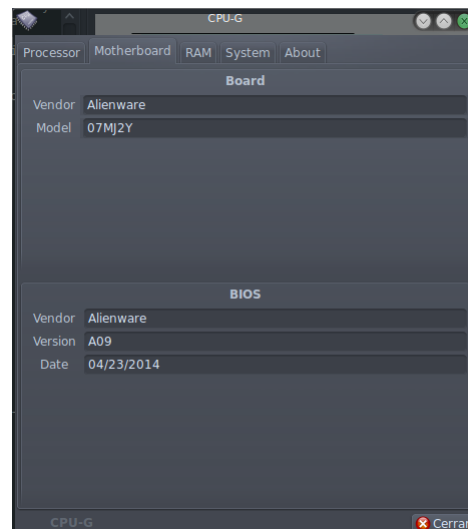


Figura 6.4: Informacion de la placa base de CPU-G

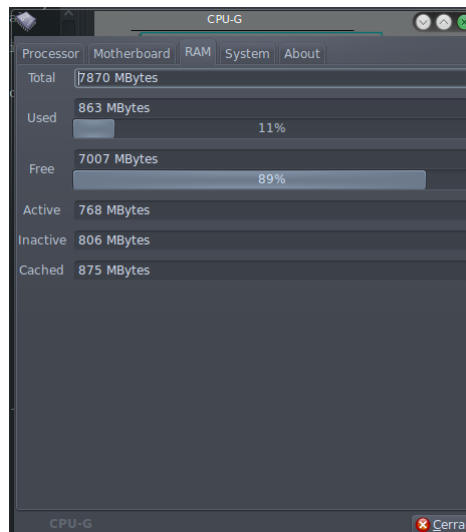


Figura 6.5: Informacion de RAM de CPU-G

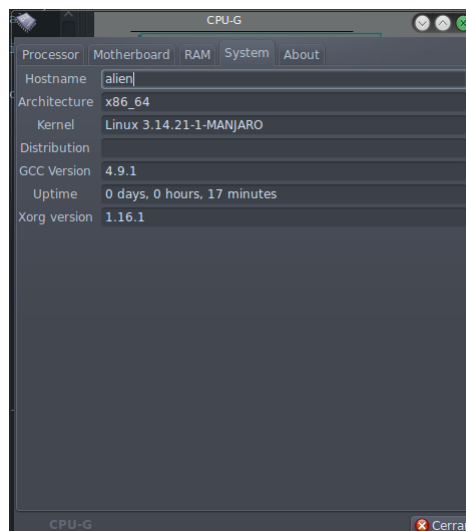


Figura 6.6: Informacion adicional del sistema en CPU-G

7. Cuestión 7

7.1. Visite la web del proyecto y acceda a la demo que proporcionan (<http://demo.munin-monitoring.org/>) donde se muestra cómo monitorizan un servidor. Monitoree varios parámetros y haga capturas de pantalla de lo que está mostrando comentando qué observa.

La demo⁴ nos permite ver una gran cantidad de graficos relativos a muchos parámetros. Aquí se han elegido unos pocos para mostrar cómo son las gráficas en munin. Cada monitorización nos muestra dos gráficas, una diaria y otra semanal. Aparte, también muestran valores actuales, mínimos, medios, y máximos.

En la sección de disco podemos ver las entradas y salidas por dispositivo, en el caso de este servidor tenemos solo un disco así que solo nos muestra una línea, en caso de tener mas nos saldrían varias.

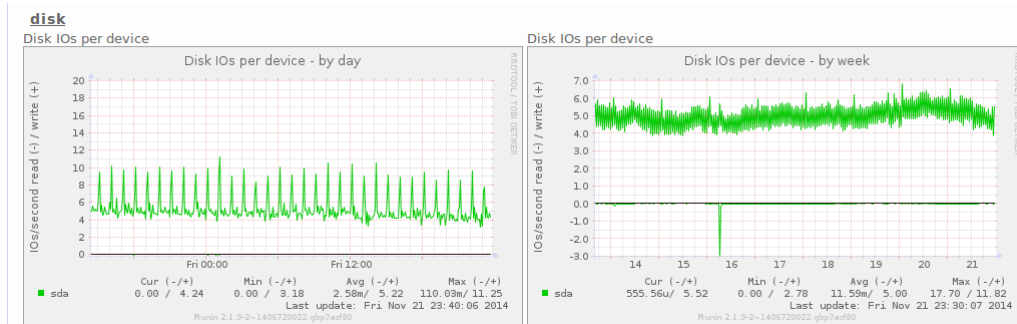


Figura 7.1: Entrada/salida por dispositivo

Tenemos también una sección de apache, en donde podemos ver entre otras cosas, los accesos por segundo a apache.

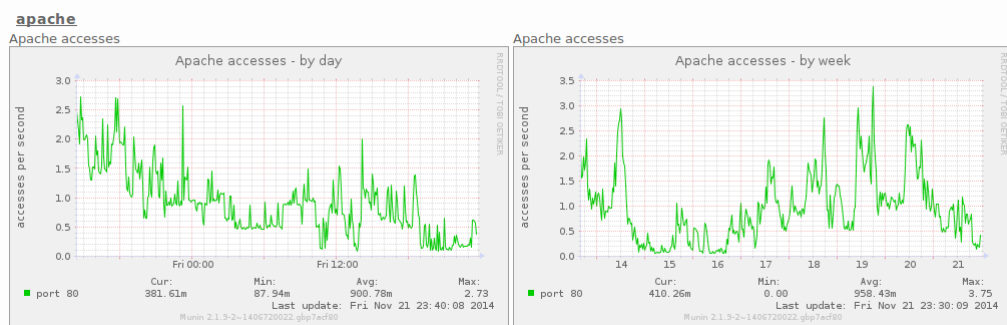


Figura 7.2: Accesos a apache

⁴<http://demo.munin-monitoring.org/munin-monitoring.org/demo.munin-monitoring.org/index.html>

Podemos por supuesto monitorizar la red, y tenemos una gráfica que nos muestra las transferencias por segundo en cada interfaz, en el caso de la siguiente gráfica, de la eth0:

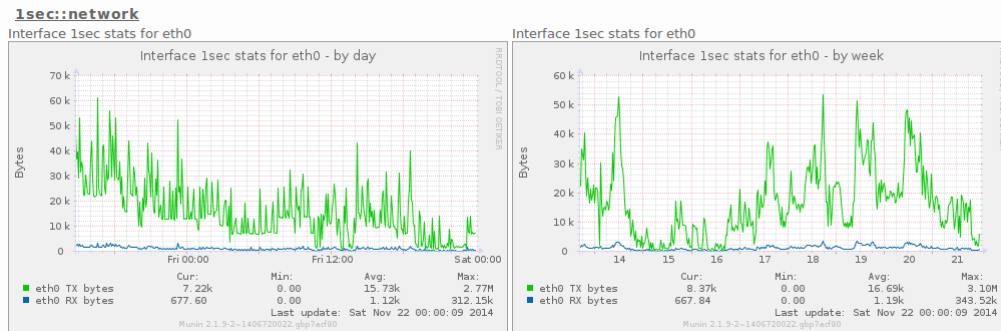


Figura 7.3: Transferencias por la interfaz eth0

También podemos monitorizar la carga del sistema, como en la siguiente gráfica que nos muestra el número de threads por segundo:

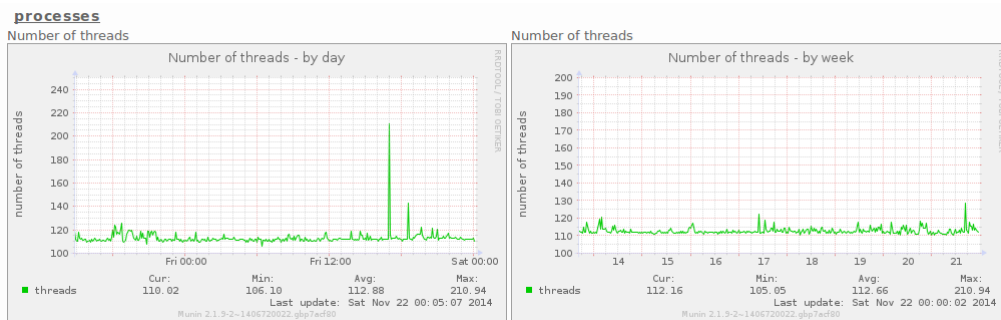


Figura 7.4: Threads activos

8. Cuestión 8

8.1. Escriba un breve resumen sobre alguno de los artículos donde se muestra el uso de strace o busque otro y coméntelo.

En el artículo que he leído⁵ tenemos a un administrador que recibe el cargo de un servidor SVN y éste se cuelga al conectarse los clientes de la oficina. Como no sabe por qué falla, sustituye el ejecutable que los clientes utilizan para conectarse por un script que llama al auténtico ejecutable con `strace`. Si el ejecutable se llama `svnserve`, lo que hace es renombrarlo a `svnserve.real` y crear el script como `svnserve` tal que así:

⁵https://www.debian-administration.org/article/352/Using_strace_to_debug_application_errors

```

1 #!/bin/sh
2 #
3 # temporary wrapper for svnserve, log output to /tmp
4 #
5 strace -o/tmp/svnserve.$$ /usr/bin/svnserve.real $*

```

De esta forma, cada vez que un cliente se conecta al servidor se guarda un log en /tmp creado por strace, el cual contiene todas las llamadas al sistema.

Después de no mas de un par de conexiones al servidor, pudo ver en los logs que el fallo estaba en una llamada al sistema read, despues de intentar abrir un descriptor para el archivo especial /dev/random:

```

1 open("/dev/random", O_RDONLY)          = 5
2 read(5,

```

Él lo soluciona cambiando /dev/random por /dev/urandom, pero obviando el cómo resuelve el problema, lo realmente interesante del artículo es la rapidez con la que lo identifica gracias a strace, que le permite ver exactamente en qué llamada se esta colgando el ejecutable.

9. Cuestion 9

9.1. Acceda a la consola mysql (o a través de phpMyAdmin) y muestre el resultado de mostrar el "profile" de una consulta (la creación de la BD y la consulta la puede hacer libremente).

Crearemos una base de datos llamada "practica_ISE", y una tabla de prueba llamada "ejercicio9_test". Vamos a la consola de comandos de la phpMyAdmin(Figura 9.1 y escribimos los comandos necesarios para activar el profiling, y hacer alguna operacion que registre datos en dicha funcionalidad⁶.

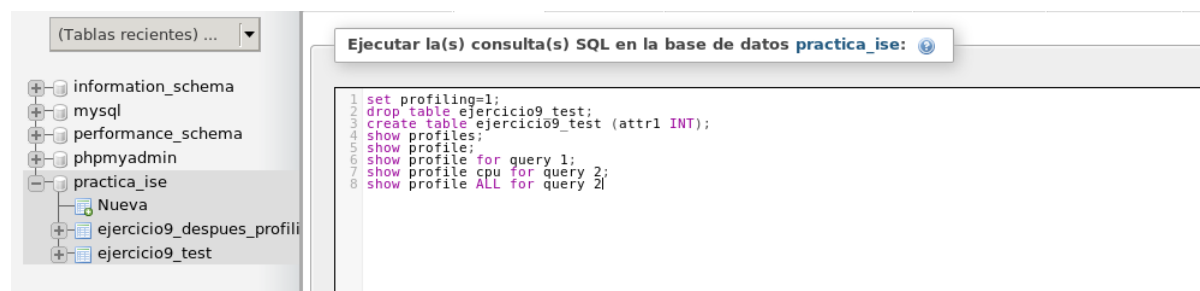


Figura 9.1: Consola de comandos de PHPMYAdmin

Comenzamos borrando y volviendo a crear la tabla anteriormente mencionada, y despues mostramos las consultas registradas con "SHOW profiles"(Figura 9.2).

⁶<http://dev.mysql.com/doc/refman/5.0/en/show-profiles.html>

Su consulta se ejecutó con éxito

SHOW profiles [\[Editar \]](#) [\[Crear código PHP \]](#)

Query_ID	Duration	Query
1	0.00405675	drop table ejercicio9_test
2	0.00417525	create table ejercicio9_test (attr1 INT)
3	0.00001825	SHOW INDEXES FROM `practica_ise`.
4	0.00007725	SHOW TABLES FROM `practica_ise`
5	0.00008700	SHOW DATABASES

Figura 9.2: SHOW profiles, que nos muestra todas las consultas registradas.

Con "SHOW profile" mostramos el registro de la última consulta realizada(fig. 9.2) en nuestro caso la 5 (ver figura 9.2), "SHOW DATABASES".

SHOW profile [\[Editar \]](#) [\[Crear código PHP \]](#)

Status	Duration
starting	0.000009
Opening tables	0.000011
System lock	0.000002
init	0.000002
optimizing	0.000001
statistics	0.000002
preparing	0.000002
executing	0.000041
Sending data	0.000005
end	0.000002
query end	0.000001
closing tables	0.000001
removing tmp table	0.000002
closing tables	0.000001
freeing items	0.000003
logging slow query	0.000001
cleaning up	0.000001

Figura 9.3: SHOW profile, que muestra el profiling de la consulta mas reciente.

Para especificar sobre qué consulta queremos ver el registro, hemos de añadir a "SHOW profile" el operador "FOR QUERY n" (ver figura 9.4), donde n es el numero de consulta que queremos ver según aparece en "SHOW profiles"

Si queremos ver un tipo concreto de registro se lo podemos especificar antes del operador "FOR", como en el caso de la figura 9.5, donde especificamos los datos de uso de CPU.

Si queremos mostrar el **registro completo** le especificaremos ALL (figura 9.6)

Su consulta se ejecutó con éxito

SHOW profile **FOR** query1 [\[Editar \]](#) [\[Crear código PHP \]](#)

Status	Duration
starting	0.000019
checking permissions	0.003994
Waiting for query cache lock	0.000015
checking permissions	0.000004
query end	0.000003
closing tables	0.000006
freeing items	0.000013
logging slow query	0.000001
cleaning up	0.000002

Figura 9.4: SHOW profile, con “for query n” nos muestra el profiling relativo a esa consulta n

SHOW profile cpu **FOR** query2 [\[Editar \]](#) [\[Crear código PHP \]](#)

Status	Duration	CPU_user	CPU_system
starting	0.000022	0.000000	0.000000
checking permissions	0.000004	0.000000	0.000000
Opening tables	0.000017	0.000000	0.000000
System lock	0.000009	0.000000	0.000000
creating table	0.004109	0.000000	0.000000
After create	0.000003	0.000000	0.000000
query end	0.000001	0.000000	0.000000
closing tables	0.000003	0.000000	0.000000
freeing items	0.000006	0.000000	0.000000
logging slow query	0.000001	0.000000	0.000000
cleaning up	0.000001	0.000000	0.000000

Figura 9.5: SHOW profile con un especificador.

SHOW profile ALL FOR query2 [Editar] [Crear código PHP]											
Status	Duration	CPU_user	CPU_system	Context_voluntary	Context_involuntary	Block_ops_in	Block_ops_out	Messages_sent	Messages_received	Page_faults_major	Page_faults
starting	0.000022	0.000000	0.000000	0	0	0	0	0	0	0	0
checking permissions	0.000004	0.000000	0.000000	0	0	0	0	0	0	0	0
Opening tables	0.000017	0.000000	0.000000	0	0	0	0	0	0	0	0
System lock	0.000009	0.000000	0.000000	0	0	0	0	0	0	0	0
creating table	0.004109	0.000000	0.000000	4	0	0	40	0	0	0	0
After create	0.000003	0.000000	0.000000	0	0	0	0	0	0	0	0
query end	0.000001	0.000000	0.000000	0	0	0	0	0	0	0	0
closing tables	0.000003	0.000000	0.000000	0	0	0	0	0	0	0	0
freeing items	0.000006	0.000000	0.000000	0	0	0	0	0	0	0	0
logging slow query	0.000001	0.000000	0.000000	0	0	0	0	0	0	0	0
cleaning up	0.000001	0.000000	0.000000	0	0	0	0	0	0	0	0

Figura 9.6: SHOW profile con el especificador ALL

10. Cuestion opcional 1

10.1. Indique qué comandos ha utilizado para realizarlo así como capturas de pantalla del proceso de reconstrucción del RAID.

Primero compruebo cómo está el array viendo el archivo `/proc/mdstat`

```
usuario@ubuntu:~$ cat /proc/mdstat
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sdb1[1] sda1[0]
      8382400 blocks super 1.2 [2/2] [UU]

unused devices: <none>
usuario@ubuntu:~$
```

Figura 10.1: `/proc/mdstat` antes de fallar ningún disco

VirtualBox no nos permite extraer el disco con la máquina encendida, así que tenemos que apagarla, y sustituir el disco antiguo (figura 10.2) por uno nuevo vacío (figura 10.3).

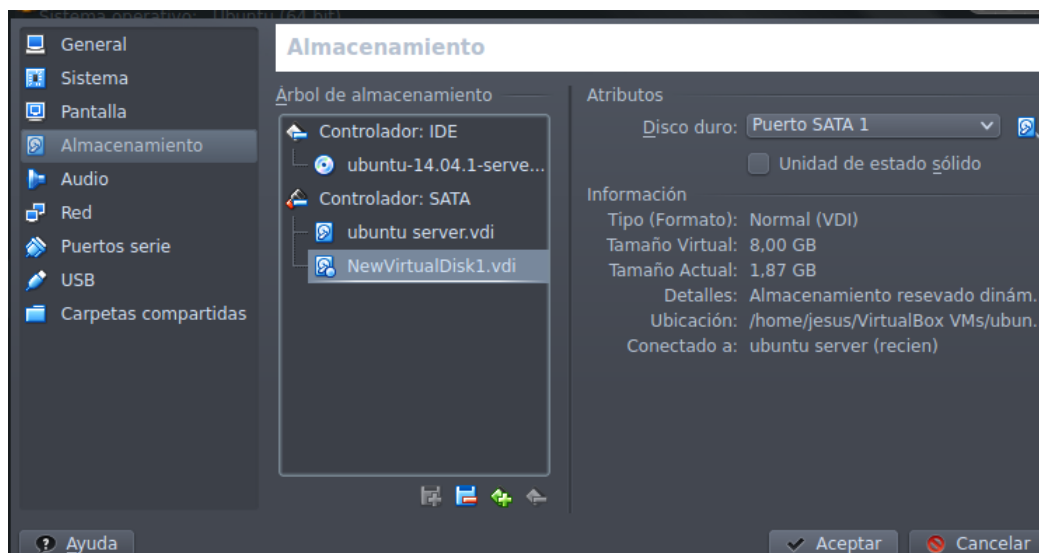


Figura 10.2: RAID con disco antiguo

Al arrancar vamos a tener un error y nos aparecerá el prompt del `initramfs` (figura 10.4). Siguiendo las instrucciones que Alberto Guillén ha colgado en SWAD, introducimos los siguientes comandos:

```
1 (initramfs) mdadm -R /dev/md0
2 (initramfs) exit
```

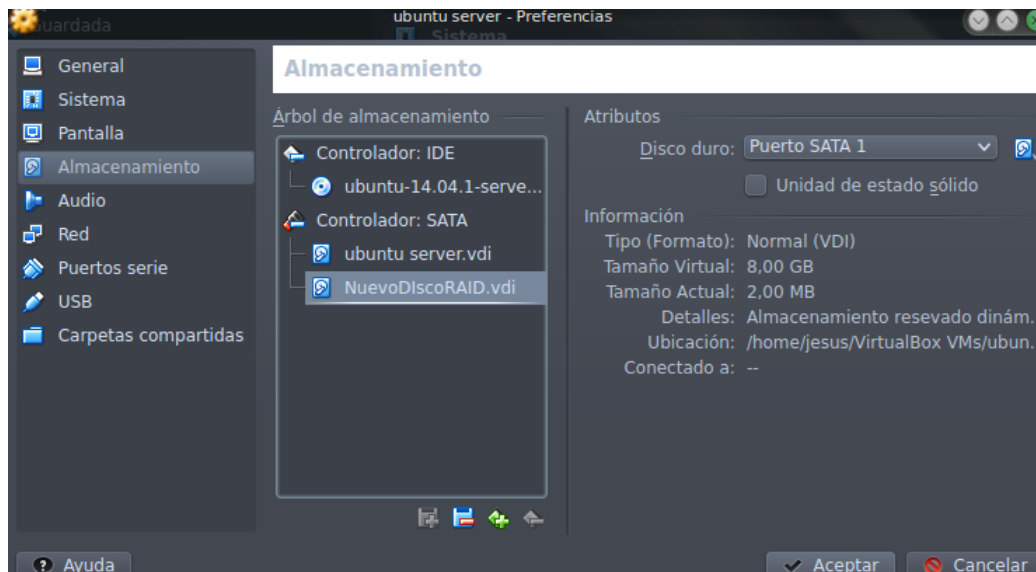


Figura 10.3: RAID con el nuevo disco

Con ello podemos continuar el arranque normalmente, “demostrando que hemos sobrevivido sin un disco gracias a haber configurado un RAID1” citando al propio Guillén.

```
BusyBox v1.21.1 (Ubuntu 1:1.21.0-1ubuntu1) built-in shell (ash)
Enter 'help' for a list of built-in commands.

(initramfs) _
```

Figura 10.4: Propmt de initramfs

Una vez arrancado el sistema volvemos a comprobar el `/proc/mdstat` y veremos lo siguiente:

```
root@ubuntu:/home/usuario# cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid10]
md0 : active raid1 sda1[0]
      8382400 blocks super 1.2 [2/1] [U_]

unused devices: <none>
root@ubuntu:/home/usuario# _
```

Figura 10.5: `/proc/mdstat` con un disco dañado o cambiado

Como podemos ver, nos aparece “[U_]”, lo que significa que el segundo disco o bien ha fallado, o bien ha sido sustituido (que para el caso es lo mismo: los discos no están sincronizados). Si queremos mas detalle tambien podemos usarlo con el comando


```
1 # mdadm --detail /dev/md0
```

La salida la podemos ver en la figura 10.6

```
root@ubuntu:/home/usuario# sudo mdadm --detail /dev/md0
/dev/md0:
    Version : 1.2
  Creation Time : Sat Oct 18 19:02:57 2014
    Raid Level : raid1
    Array Size : 8382400 (7.99 GiB 8.58 GB)
  Used Dev Size : 8382400 (7.99 GiB 8.58 GB)
    Raid Devices : 2
    Total Devices : 1
 Persistence : Superblock is persistent

 Update Time : Fri Nov 21 13:09:31 2014
   State : clean, degraded
Active Devices : 1
Working Devices : 1
Failed Devices : 0
Spare Devices : 0

    Name : ubuntu:0 (local to host ubuntu)
   UUID : f45ef483:8f2d5f6f:e45949e9:f8400c21
  Events : 147

   Number Major Minor RaidDevice State
     0       8       1        0   active sync  /dev/sda1
     1       0       0        1   removed

root@ubuntu:/home/usuario#
```

Figura 10.6: mdadm con el modificador -detail

Procedemos a eliminar la particion perteneciente al disco eliminado (ya lo eliminamos antes, pero por si acaso) con el comando:

```
1 # mdadm --manage /dev/md0 --remove detached
```

Ahora procedemos a clonar la tabla de particiones del disco que está bien, al nuevo disco (deben tener el mismo tamaño, muy importante). Para ello vamos a usar los siguientes comandos:

```
1 # sfdisk -d /dev/sda > sda_particiones.txt
2 # sfdisk /dev/sdb < sda_particiones.txt
```

Una vez clonada la tabla de particiones, añadimos la nueva particion al raid con:

```
1 # mdadm --detail /dev/md0
```

El nuevo disco se añade al raid y podemos comprobar el progreso de clonado de dos formas, una con la mencionada en el guión(figura 10.7, y otra con la anteriormente mostrada mediante mdstat (figura 10.8), extraída de la referencia usada para elaborar esta pregunta⁷.

⁷<http://www.eslomas.com/2012/06/reconstruir-dispositivo-raid1-tras-fallar-un-disco-duro/>

```

root@ubuntu:/home/usuario# mdadm --detail /dev/md0
/dev/md0:
    Version : 1.2
  Creation Time : Sat Oct 18 19:02:57 2014
    Raid Level : raid1
    Array Size : 8382400 (7.99 GiB 8.58 GB)
  Used Dev Size : 8382400 (7.99 GiB 8.58 GB)
    Raid Devices : 2
  Total Devices : 2
 Persistence : Superblock is persistent

    Update Time : Fri Nov 21 13:50:57 2014
      State : clean, degraded, recovering
  Active Devices : 1
 Working Devices : 2
 Failed Devices : 0
  Spare Devices : 1

Rebuild Status : 1% complete

        Name : ubuntu:0 (local to host ubuntu)
        UUID : f45ef483:8f2d5f6f:e45949e9:f8400c21
        Events : 461

   Number   Major   Minor   RaidDevice State
     0         8         1         0     active sync   /dev/sda1
     2         8        17         1     spare rebuilding /dev/sdb1
root@ubuntu:/home/usuario#

```

Figura 10.7: Proceso de reconstrucción consultado con mdstat

```

Cada 2,0s: cat /proc/mdstat                               Fri Nov 21 13:51:49 2014
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sdb1[2] sda1[0]
      8382400 blocks super 1.2 [2/1] [U_]
      [=>.....] recovery = 10.1% (851648/8382400) finish=9.3min speed=13436K/sec
unused devices: <none>

```

Figura 10.8: Proceso de reconstrucción consultado con watch y /proc/mdstat

11. Cuestion opcional 2

11.1. Instale Nagios en su sistema (el que prefiera) documentando el proceso y muestre el resultado de la monitorización de su sistema comentando qué aparece.

Para instalar Nagios en CentOS⁸ tenemos que instalar antes todas las dependencias:

```
1 # yum install -y httpd php gcc glibc glibc-common gd gd-devel make net-snmp
```

Tambien es necesario crear un usuario y un grupo para Nagios:

```
1 # useradd nagios
2 # groupadd nagcmd
```

Es necesario añadir tanto al usuario nagios como al usuario apache al grupo nagcmd:

```
1 # usermod -G nagcmd nagios
2 # usermod -G nagcmd apache
```

Descargamos las fuentes de Nagios y sus plugins en la carpeta que queramos:

```
1 # wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.1.tar.gz
2 # wget https://www.nagios-plugins.org/download/nagios-plugins-1.5.tar.gz
```

Descomprimos las dos carpetas como prefiramos y entramos a la carpeta del core para configurarlo, y para ello tenemos que pasarle al script de configuracion el nombre del grupo que hemos creado tal que así:

```
1 # ./configure --with-command-group=nagcmd
```

Al final de la configuracion nos mostrará un resumen como el mostrado en la figura 11.1. Como todo parece estar bien, seguimos las instrucciones y compilamos con `make all`.

```
1 # make all
2 # make install
```

Tras acabar la instalacion (figura 11.2), nos muestra algunos comandos mas que deberemos ejecutar para completar la instalacion. Ejecutamos los tres comandos y habremos acabado de instalar el núcleo de Nagios..

Para instalar la interfaz web tenemos que ejecutar el siguiente comando:

```
1 # make install-webconf
```

Esto crea un usuario “nagiosadmin” para acceder a dicha interfaz. Es necesario que le demos una contraseña (por ejemplo nagiospass) y para ello usamos el comando que se muestra a continuación:

⁸<http://www.linux-party.com/index.php/106-nagios/9001-instalar-nagios-4-0-1-en-rhel-centos-6>

```

*** Configuration summary for nagios 4.0.1 10-15-2013 ***:

General Options:
-----
Nagios executable: nagios
Nagios user/group: nagios,nagios
Command user/group: nagios,nagcmd
Event Broker: yes
Install ${prefix}: /usr/local/nagios
Install ${includedir}: /usr/local/nagios/include/nagios
Lock file: ${prefix}/var/nagios.lock
Check result directory: ${prefix}/var/spool/checkresults
Init directory: /etc/rc.d/init.d
Apache conf.d directory: /etc/httpd/conf.d
Mail program: /bin/mail
Host OS: linux-gnu

Web Interface Options:
-----
HTML URL: http://localhost/nagios/
CGI URL: http://localhost/nagios/cgi-bin/
Traceroute (used by WAP): /bin/traceroute

Review the options above for accuracy. If they look okay,
type 'make all' to compile the main program and CGIs.

```

Figura 11.1: Resumen de la configuracion de Nagios

```

*** Main program, CGIs and HTML files installed ***

You can continue with installing Nagios as follows (type 'make'
without any arguments for a list of all possible options):

make install-init
- This installs the init script in /etc/rc.d/init.d

make install-commandmode
- This installs and configures permissions on the
  directory for holding the external command file

make install-config
- This installs sample config files in /usr/local/nagios/etc

make[1]: se sale del directorio `/home/usuario/nagios/nagios-4.0.1'
[root@centos-server nagios-4.0.1]# █

```

Figura 11.2: Salida de make install

```
1 # htpasswd -s -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

Para que la configuración de la interfaz surta efecto tenemos que reiniciar apache:

```
1 # systemctl restart httpd.service
```

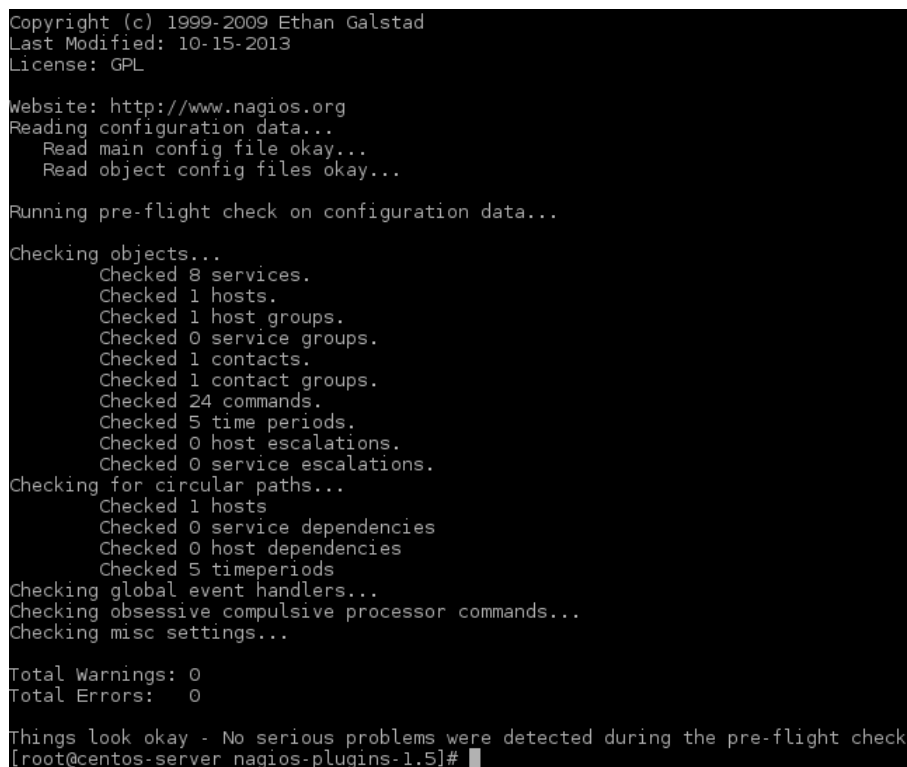
Si queremos instalar los plugins, descomprimos de igual forma el fichero que descargamos antes, configuramos la compilación (especificando el usuario), y compilamos:

```
1 # ./configure --with-nagios-user=nagios --with-nagios-group=nagios
2 # make
3 # make install
```

También podemos verificar la configuración si lo deseamos:

```
1 # /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

Deberá aparecer algo como lo mostrado en la figura 11.3

A terminal window showing the output of the Nagios configuration verification command. The output includes copyright information, website URL, and a detailed pre-flight check of the configuration. The check lists various components like services, hosts, groups, commands, and time periods, all of which are checked successfully. The final result is 'Things look okay - No serious problems were detected during the pre-flight check'.

```
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 10-15-2013
License: GPL

Website: http://www.nagios.org
Reading configuration data...
  Read main config file okay...
  Read object config files okay...

Running pre-flight check on configuration data...

Checking objects...
  Checked 8 services.
  Checked 1 hosts.
  Checked 1 host groups.
  Checked 0 service groups.
  Checked 1 contacts.
  Checked 1 contact groups.
  Checked 24 commands.
  Checked 5 time periods.
  Checked 0 host escalations.
  Checked 0 service escalations.
Checking for circular paths...
  Checked 1 hosts
  Checked 0 service dependencies
  Checked 0 host dependencies
  Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
[root@centos-server nagios-plugins-1.5]#
```

Figura 11.3: Comprobación de la configuración de Nagios

Finalmente, para que Nagios funcione tras reiniciar habilitamos el servicio:

```

1 # chkconfig --add nagios
2 # chkconfig --level 35 nagios on
3 # systemctl start nagios.service

```

Y por fin, podemos acceder a Nagios a través de nuestro navegador con la dirección `localhost/nagios`, con el usuario `nagiosadmin` y la contraseña que le pusieramos. La pantalla que nos da la bienvenida la podemos ver en la figura 11.4

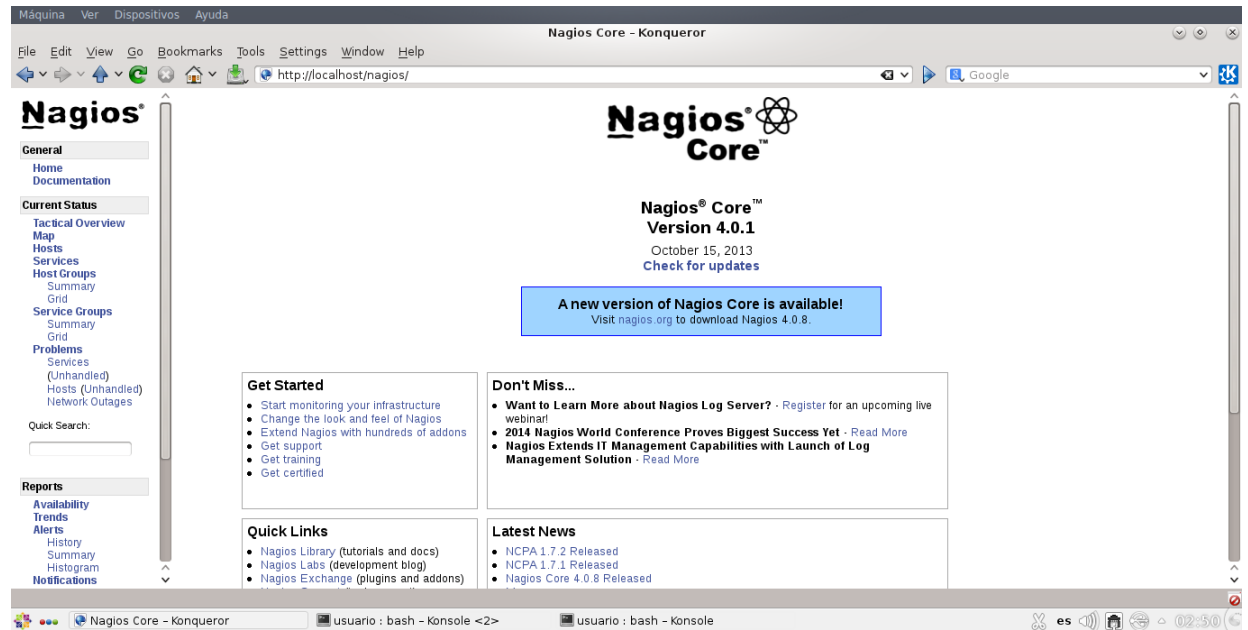


Figura 11.4: Pantalla inicial de Nagios

Para ver un resumen de las notificaciones (figura 11.5 pinchamos en “Tactical Overview”. Como podemos ver, nos muestra un resumen de los hosts conectados al servidor, un resumen de los servicios activos, y las características de monitorización activadas.

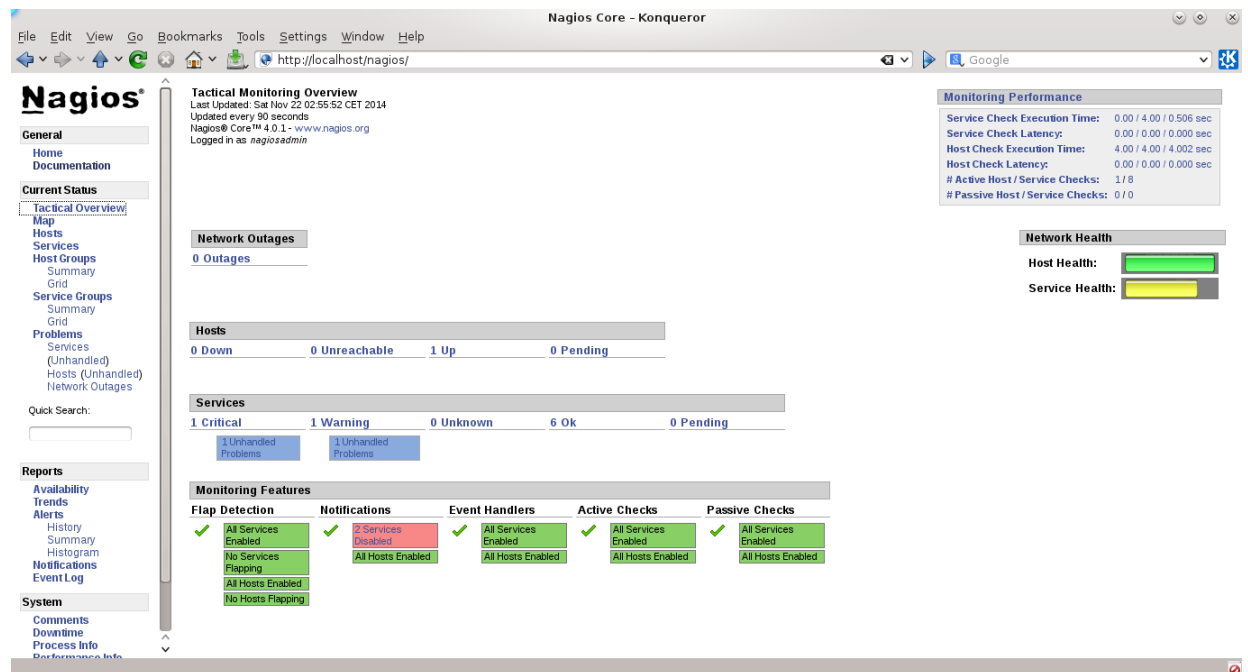


Figura 11.5: Resumen del sistema

12. Cuestión opcional 3

12.1. Haga lo mismo que con Munin

Entramos a la web de Ganglia y entramos a la demo que allí proporcionan ⁹ Ahi podemos ver la carga de trabajo agrupada a mayor o menor nivel. Ganglia tambien nos permite cambiar el periodo de monitorizacion en la parte de arriba. En la figura 12.1 podemos ver la carga de trabajo, uso de memoria, y de red del grid de wikimedia. Ademas vemos a la izquierda el numero de CPUs de las que se compone, los hosts que tiene conectados, y los que tiene desconectados. Si queremos consultar un cluster concreto dentro del grid, podemos elegirlo tambien en la parte superior, y nos mostrará la misma informacion pero solo referente a dicho clúster, como en el ejemplo de la figura 12.2. Por último nos permite monitorizar a nivel de host, como se ve en la figura 12.3

⁹<http://ganglia.wikimedia.org/latest/>

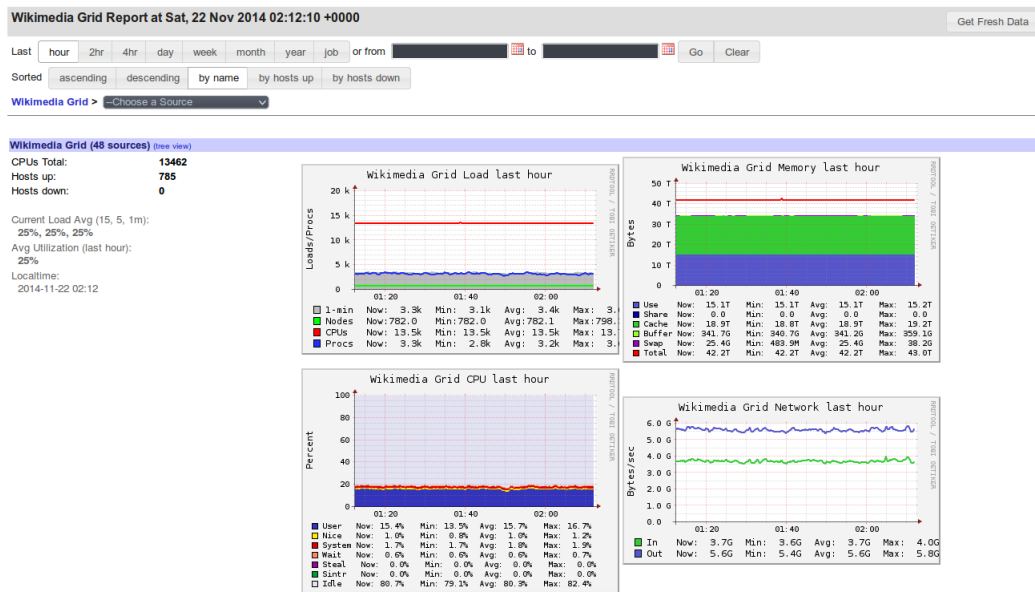


Figura 12.1: Carga de trabajo del grid de Wikimedia completo

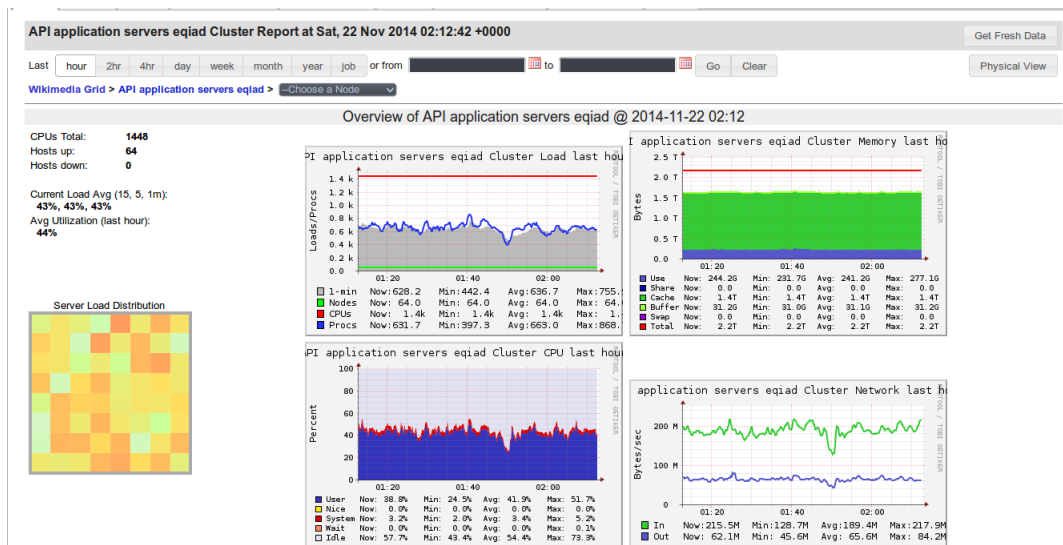


Figura 12.2: Carga de trabajo de un cluster del grid

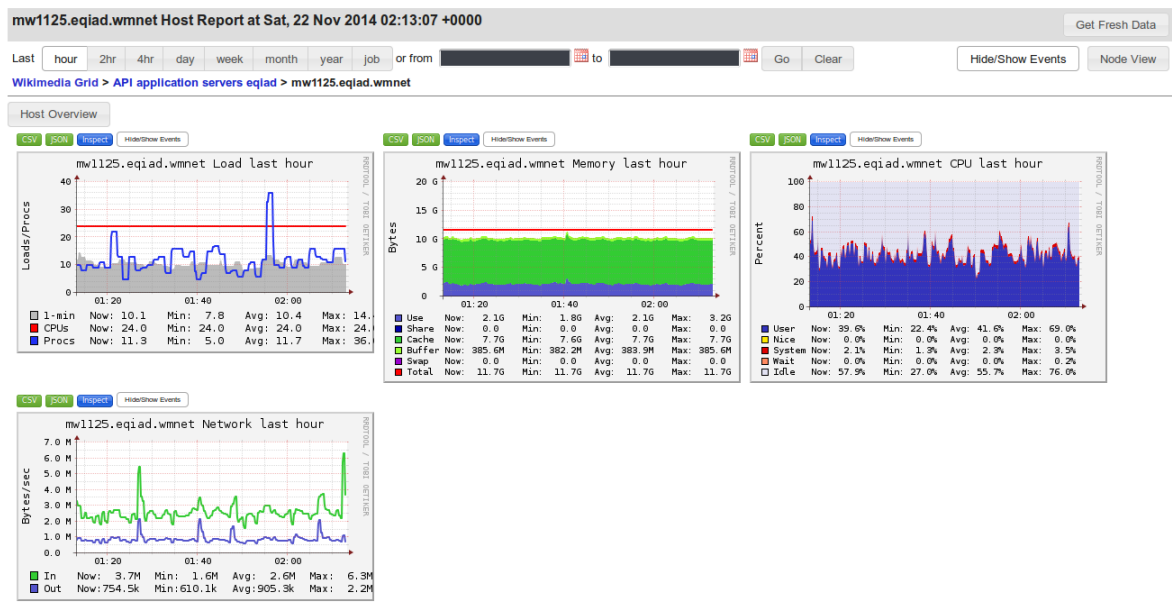


Figura 12.3: Carga de trabajo de un nodo de un cluster