

INGENIERÍA DE SERVIDORES (2016-2017)
SUBGRUPO A1
GRADO EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

Práctica 3: Monitorización de servicios

Mario Rodríguez Ruiz

9 de diciembre de 2016

Índice

1	Cuestión 1	5
1.1	¿Qué archivo le permite ver qué programas se han instalado con el gestor de paquetes?	5
1.2	¿Qué significan las terminaciones . 1.gz o .2.gz de los archivos en ese directorio?	5
2	Cuestión 2	7
2.1	¿Qué archivo ha de modificar para programar una tarea?	7
2.2	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)	7
3	Cuestión 3	8
3.1	Pruebe a ejecutar el comando, conectar un dispositivo USB y vuelva a ejecutar el comando. Copie y pegue la salida del comando.	8
3.2	Comente qué observa en la información mostrada.	8
4	Cuestión 4	9
4.1	Ejecute el monitor de "System Performance" y muestre el resultado. Incluya capturas de pantalla comentando la información que aparece.	9
5	Cuestión 5	12
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	12
6	Cuestión 6	19
6.1	Visite la web del proyecto y acceda a la demo que proporcionan (http://demo.munin-monitoring.org/) donde se muestra cómo monitorizar un servidor. Monitoree varios parámetros y haga capturas de pantalla de lo que está mostrando comentando qué observa.	19
7	Cuestión 7	23
7.1	Escriba un breve resumen sobre alguno de los artículos donde se muestra el uso de strace o busque otro y coméntelo	23
8	Cuestión 8	24
8.1	Escriba un script en Python o PHP	24
8.2	Analice el comportamiento del script usando el profiler presentado	25
9	Cuestión 9	30
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)	30

10 Cuestión opcional 2	34
10.1 Instale Nagios en su sistema (el que prefiera) documentando el proceso y muestre el resultado de la monitorización de su sistema comentando qué aparece	34

Índice de figuras

1.1. Contenido del fichero /var/log/yum.log en CentOS	5
1.2. Contenido del fichero /etc/logrotate.conf en CentOS	6
2.1. Contenido del fichero /etc/crontab en CentOS	7
3.1. Ejecución del comando dmesg antes	8
3.2. Ejecución del comando dmesg después	8
4.1. Ventana principal del Monitor de rendimiento de Windows	9
4.2. Ejecución del System Performance	10
4.3. Ejecución del System Performance	11
4.4. Pestañas de información detallada de System Performance	11
5.1. Nuevo conjunto de recopiladores	12
5.2. Nombre y modo de creación del conjunto	13
5.3. Tipos de datos a incluir en el recopilador	13
5.4. Qué contadores de rendimiento registrar	14
5.5. Agregar contenedores	15
5.6. Contadores de rendimiento para registrar	15
5.7. Directorio de destino del informe	16
5.8. Finalizado del proceso de creación de recopiladores	16
5.9. Recopilador definido activo	17
5.10. Directorio de almacenamiento de resultados	17
5.11. Informe del recopilador en Monitor de rendimiento	18
6.1. IOs for /dev/sda [9]	19
6.2. Uso de memoria física [9]	20
6.3. Load average [10]	21
6.4. Procesos [11]	22
8.1. Prueba de ejecución de script en Python	25
8.2. Primera aproximación de ejecución del profiler	26
8.3. Presentación de los datos obtenidos con Kcachegrind	27
8.4. Primera aproximación de ejecución del profiler	28
8.5. Representación de la organización del tiempo de las llamadas	29
9.1. Creación de una nueva base de datos en Mysql	30
9.2. Creación de tablas en Mysql	31
9.3. Activación del profiler en Mysql	31
9.4. Utilización profiler en Mysql	32
9.5. Utilización profiler en Mysql	32
9.6. Utilización profiler en Mysql	33
10.1. Instalación de los paquetes necesarios para Nagios	34

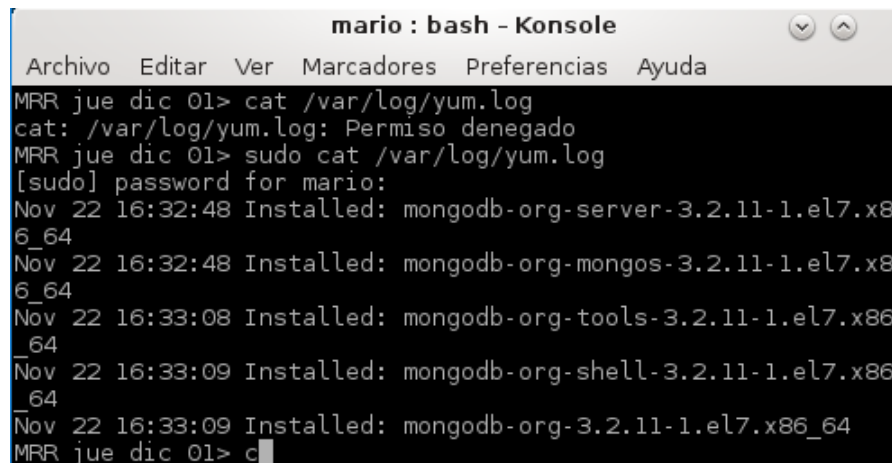
10.2. Creación de usuario y grupo para Nagios	34
10.3. Deshabilitación de SELinux	35
10.4. Descarga de Nagios y sus plugins	36
10.5. Instalación de Nagios	36
10.6. Instalación de Nagios	37
10.7. Instalación de Nagios	37
10.8. Instalación de Nagios	37
10.9. Creación de la contraseña de usuario para Nagios	38
10.10 Instalación de los plugins necesarios para Nagios	38
10.11Apertura de los puertos para Nagios	39
10.12Inicio de Nagios	39
10.13Login en Nagios	40
10.14Pantalla principal de Nagios	40
10.15Tactical Overview Nagios	41
10.16Estados de los servicios Nagios	41
10.17Informacion de rendimiento en Nagios	42

1. Cuestión 1

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

El archivo que permite ver qué programas se han instalado con el gestor de paquetes en CentOS es `/var/log/yum.log` [2]

En la Figura 1.1 se muestra el contenido actual de este fichero en la máquina CentOS.



```
mario : bash - Konsole
Archivo  Editar  Ver  Marcadores  Preferencias  Ayuda
MRR jue dic 01> cat /var/log/yum.log
cat: /var/log/yum.log: Permiso denegado
MRR jue dic 01> sudo cat /var/log/yum.log
[sudo] password for mario:
Nov 22 16:32:48 Installed: mongodb-org-server-3.2.11-1.el7.x86_64
Nov 22 16:32:48 Installed: mongodb-org-mongos-3.2.11-1.el7.x86_64
Nov 22 16:33:08 Installed: mongodb-org-tools-3.2.11-1.el7.x86_64
Nov 22 16:33:09 Installed: mongodb-org-shell-3.2.11-1.el7.x86_64
Nov 22 16:33:09 Installed: mongodb-org-3.2.11-1.el7.x86_64
MRR jue dic 01> c
```

Figura 1.1: Contenido del fichero `/var/log/yum.log` en CentOS

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

Las terminaciones `.1.gz` o `.2.gz` de los archivos en ese directorio significan que están comprimidos.

Se trata de ficheros log antiguos: como los logs van creciendo, el sistema los comprime y les aplica una rotación, poniéndole la extensión anteriormente nombrada según la antigüedad y creando uno nuevo vacío. [4] De esta forma se evita que el almacenamiento sea desbordado a causa de los logs.

La utilidad que se encarga de realizar este proceso a diario es **logrotate**, cuya funcionalidad puede ser personalizada desde su fichero de configuración `/etc/logrotate.conf` que puede verse en la en la Figura 1.2

```
MRR jue dic 01> sudo cat /etc/logrotate.conf
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# use date as a suffix of the rotated file
dateext

# uncomment this if you want your log files compressed
#compress

# RPM packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own wtmp and btmp -- we'll rotate them here
/var/log/wtmp {
    monthly
    create 0664 root utmp
    minsize 1M
    rotate 1
}

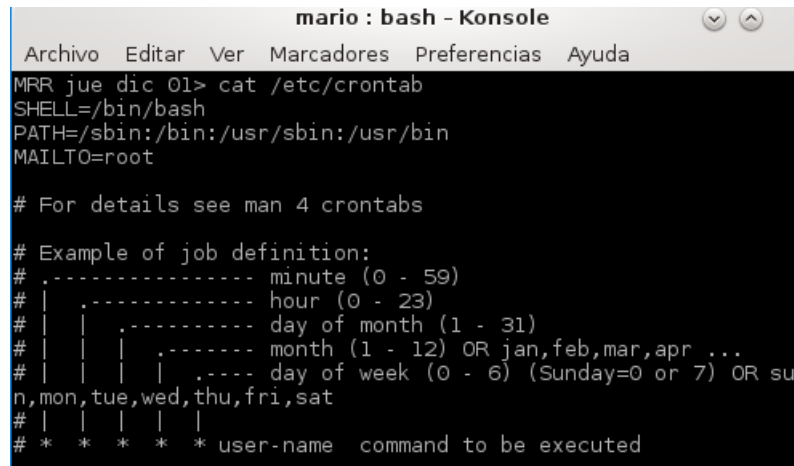
/var/log/btmp {
    missingok
    monthly
    create 0600 root utmp
    rotate 1
}
```

Figura 1.2: Contenido del fichero `/etc/logrotate.conf` en CentOS

2. Cuestión 2

2.1. ¿Qué archivo ha de modificar para programar una tarea?

El archivo que hay que modificar para programar una tarea es `/etc/crontab` [3] y su contenido actual es el que aparece en la Figura 2.1

A screenshot of a terminal window titled "mario : bash - Konsole". The window has a menu bar with "Archivo", "Editar", "Ver", "Marcadores", "Preferencias", and "Ayuda". The terminal shows the output of the command "cat /etc/crontab". The content includes system environment variables (SHELL, PATH, MAILTO), a reference to man 4 crontabs, and a detailed example of a job definition with fields for minute, hour, day of month, month, and day of week, followed by a list of days and a user-name command to be executed.

```
mario : bash - Konsole
Archivo  Editar  Ver  Marcadores  Preferencias  Ayuda
MRR jue dic 01> cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .----- day of week (0 - 6) (Sunday=0 or 7) OR su
n,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name  command to be executed
```

Figura 2.1: Contenido del fichero `/etc/crontab` en CentOS

2.2. 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`)

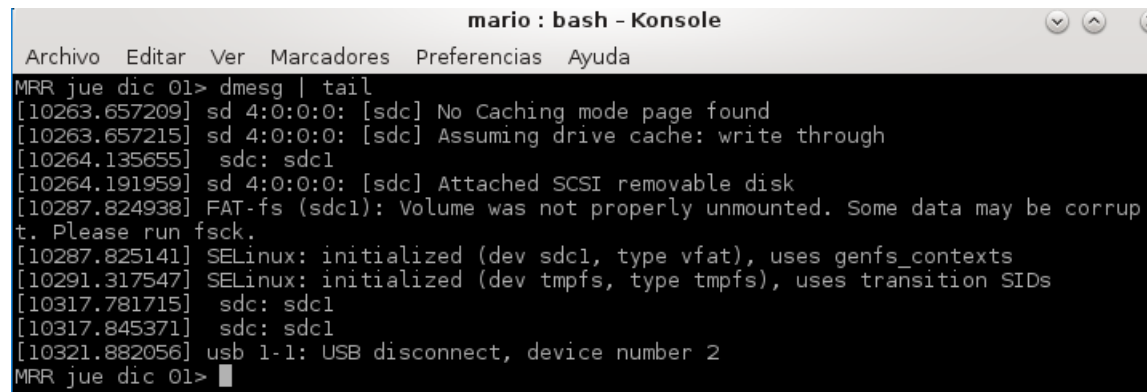
```
1 0 8 * * * user cp -r ~/codigo/* ~/seguridad/$(date +%Y%m%d)
```

Esta línea realiza la copia del directorio `~/codigo` a `~/seguridad/$fecha`, acción que será ejecutada **todos los días del año a las 08:00 horas**.

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.

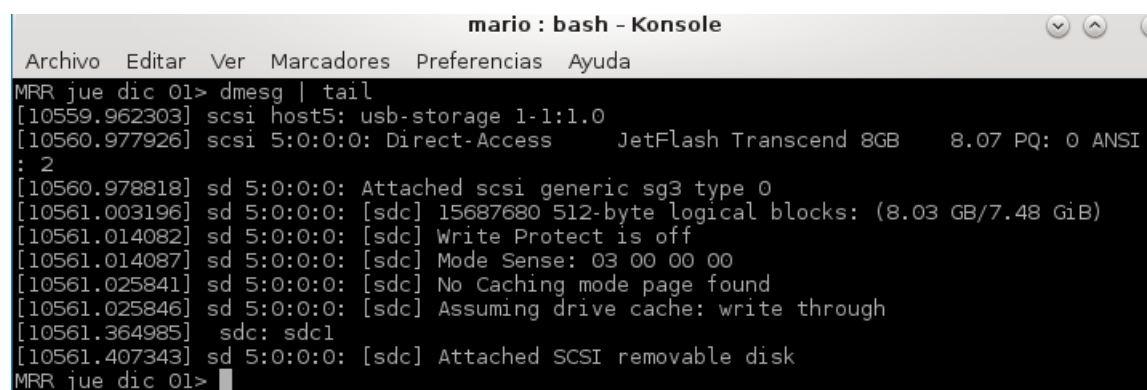
La Figura 3.1 muestra la salida del comando **antes** de la inserción del USB.



```
mario : bash - Konsole
Archivo  Editar  Ver  Marcadores  Preferencias  Ayuda
MRR jue dic 01> dmesg | tail
[10263.657209] sd 4:0:0:0: [sdc] No Caching mode page found
[10263.657215] sd 4:0:0:0: [sdc] Assuming drive cache: write through
[10264.135655] sdc: sdc1
[10264.191959] sd 4:0:0:0: [sdc] Attached SCSI removable disk
[10287.824938] FAT-fs (sdc1): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.
[10287.825141] SELinux: initialized (dev sdc1, type vfat), uses genfs_contexts
[10291.317547] SELinux: initialized (dev tmpfs, type tmpfs), uses transition SIDs
[10317.781715] sdc: sdc1
[10317.845371] sdc: sdc1
[10321.882056] usb 1-1: USB disconnect, device number 2
MRR jue dic 01> █
```

Figura 3.1: Ejecución del comando **dmesg** antes

La Figura 3.2 muestra la salida del comando **después** de la inserción del USB.



```
mario : bash - Konsole
Archivo  Editar  Ver  Marcadores  Preferencias  Ayuda
MRR jue dic 01> dmesg | tail
[10559.962303] scsi host5: usb-storage 1-1:1.0
[10560.977926] scsi 5:0:0:0: Direct-Access          JetFlash Transcend 8GB      8.07 PQ: 0 ANSI
: 2
[10560.978818] sd 5:0:0:0: Attached scsi generic sg3 type 0
[10561.003196] sd 5:0:0:0: [sdc] 15687680 512-byte logical blocks: (8.03 GB/7.48 GiB)
[10561.014082] sd 5:0:0:0: [sdc] Write Protect is off
[10561.014087] sd 5:0:0:0: [sdc] Mode Sense: 03 00 00 00
[10561.025841] sd 5:0:0:0: [sdc] No Caching mode page found
[10561.025846] sd 5:0:0:0: [sdc] Assuming drive cache: write through
[10561.364985] sdc: sdc1
[10561.407343] sd 5:0:0:0: [sdc] Attached SCSI removable disk
MRR jue dic 01> █
```

Figura 3.2: Ejecución del comando **dmesg** después

3.2. Comente qué observa en la información mostrada.

El comando **dmesg** se ha redirigido a **tail** para que aparezcan solamente los últimos eventos que se han producido en el sistema.

La información que se muestra en la Figura 3.2 es el vínculo que se ha creado entre el sistema operativo y el USB que se acaba de insertar, así como parte importante de información de éste.

En este caso el SO le ha asignado **sdc1** al USB, el **host5** como interfaz SCSI en acceso directo. Como información, se muestran los bloques de memoria lógica que tiene así como su espacio.

También aparece si tiene la escritura protegida (en este caso no) y el formato del dispositivo (Mode Sense) en el que **03 00** significa que es un dispositivo de disco [6].

Por último, se muestra si dispone de página de modo de caché (no en este caso), cómo asume la memoria cache de la unidad (mediante escritura en este caso) y qué tipo de dispositivo es (disco extraíble adjunto SCSI)

4. Cuestión 4

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

Para ejecutar el monitor de System Performance una vez dentro del Monitor de rendimiento, hay que pulsar **Iniciar** sobre **System Performance** dentro de **Sistema** que aparece en la pestaña **Conjuntos de recopiladores** en el panel lateral izquierdo de la aplicación, tal y como muestra la Figura 4.1

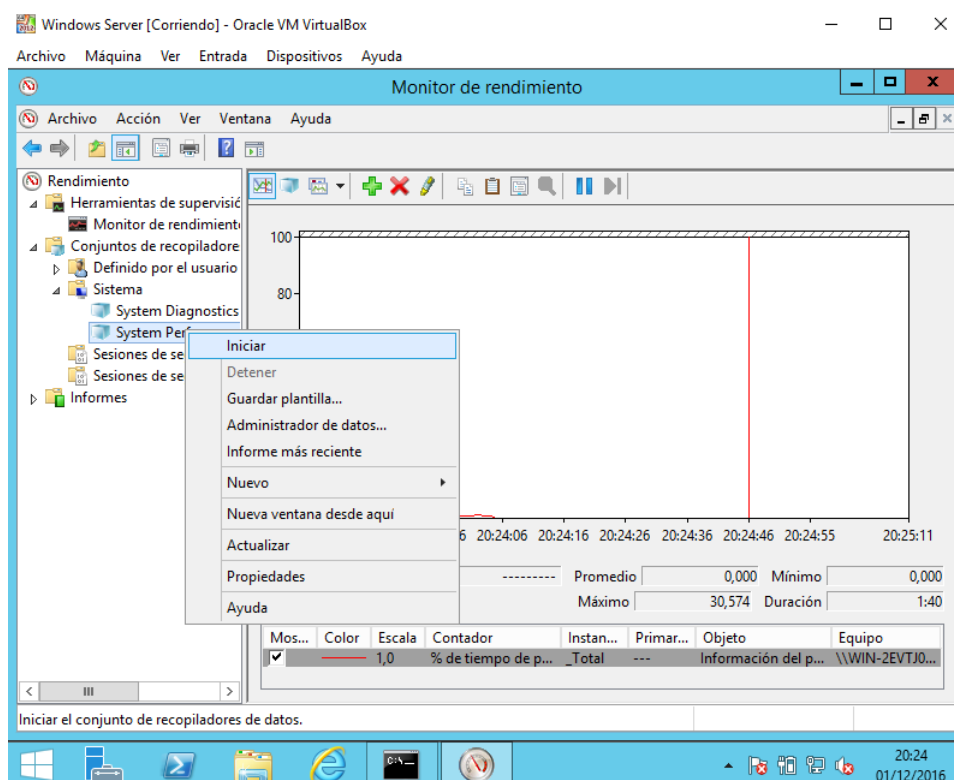


Figura 4.1: Ventana principal del Monitor de rendimiento de Windows

Cuando acaba la ejecución aparece un nuevo informe en la pestaña **Informes** en el mismo panel actual.

Para obtener la información basta con pulsar sobre éste y aparecerá un desglose con diferentes pestañas clasificadas según el tipo de información, como aparece en la Figura 4.2

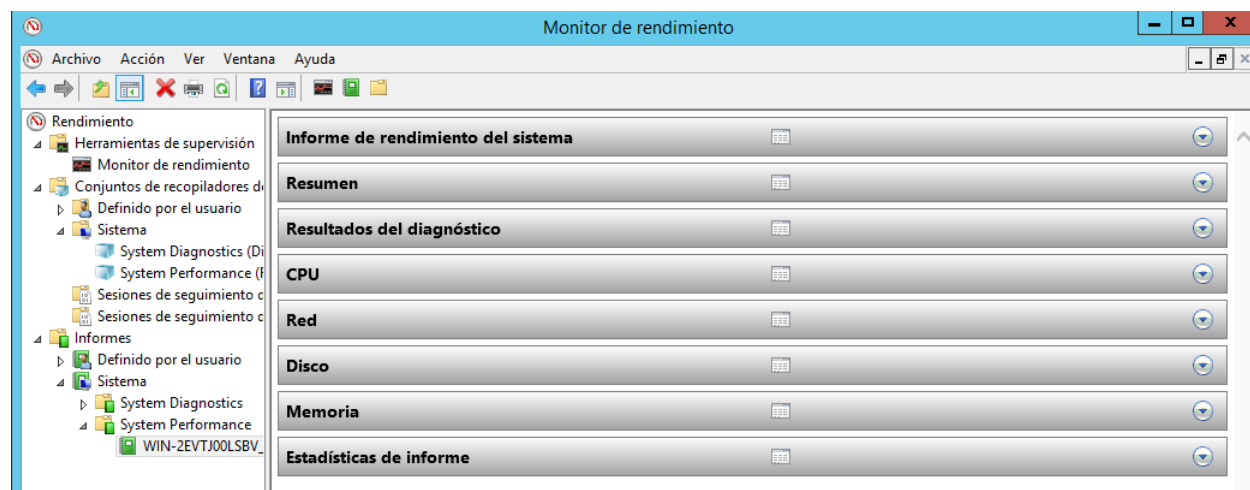


Figura 4.2: Ejecución del **System Performance**

En la Figura 4.3 aparecen abiertas algunas pestañas importantes:

- El **informe de rendimiento de sistema** en el que se muestra el equipo, la fecha y la duración de la ejecución.
- Un **resumen**, en el que aparecen diferentes evaluaciones importantes como son el uso de memoria, disco y CPU. Se trata de una información no detallada pero que puede resultar muy útil para informes rápidos.
- El **rendimiento**, en el que se presenta la información general de recursos de la CPU, Red, Disco y Memoria con un mínimo detalles.

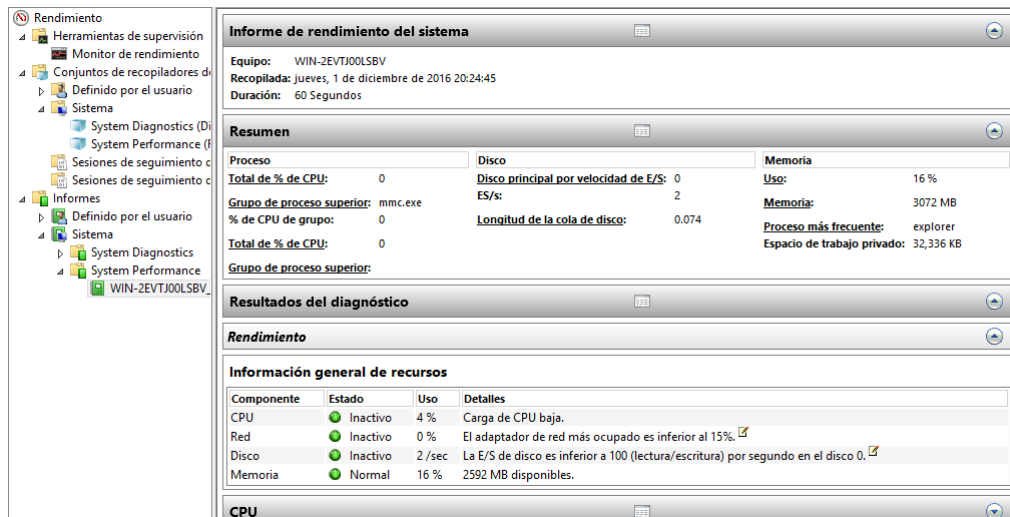


Figura 4.3: Ejecución del **System Performance**

El resto de pestañas ya sí que ofrecen unas evaluaciones bastante detalladas sobre los distintos componentes del sistema antes mencionados.

En la Figura 4.4 se ven dichas evaluaciones que, por ejemplo, permiten conocer los bytes enviados/recibidos desde una interfaz de Red específica del sistema; así como el tiempo medio de lectura/escritura en un disco duro físico, entre muchas otras.



Figura 4.4: Pestañas de información detallada de **System Performance**

5. Cuestión 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

Para comenzar lo primero es abrir la herramienta **Monitor de rendimiento**. Una vez ahí, hay que situarse en el panel lateral izquierdo de la herramienta y pulsar sobre **Conjuntos de recopiladores** → **Definido por el usuario**. Sobre éste último hacer clic con el botón derecho del ratón y, a continuación, **Nuevo** → **Conjunto de recopiladores de datos**.

La Figura 5.1 muestra dicho proceso con la ayuda de unas flechas rojas adicionales.

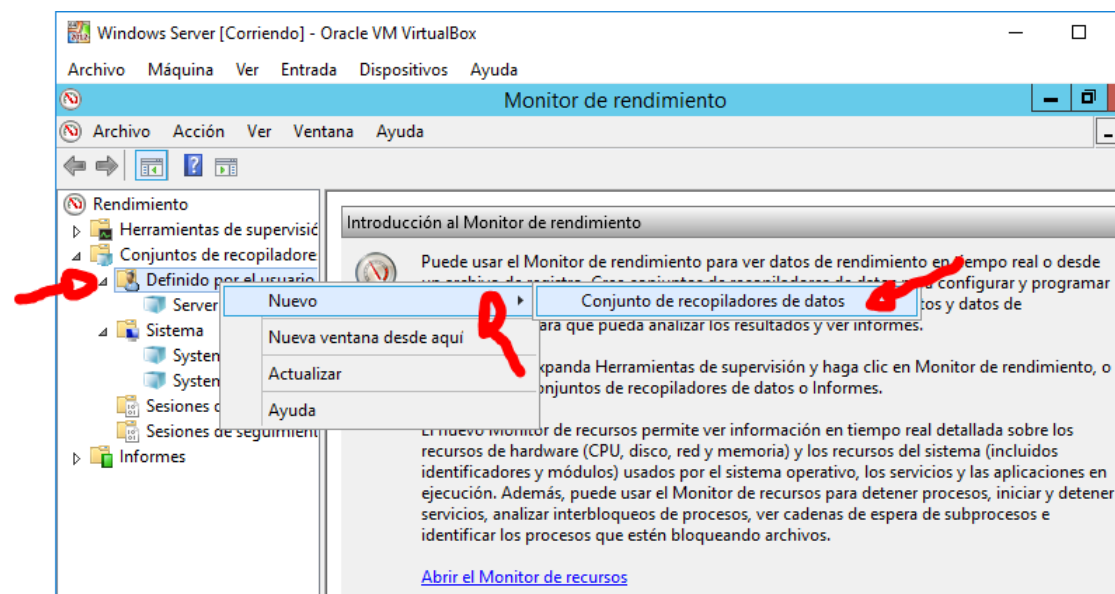


Figura 5.1: Nuevo conjunto de recopiladores

La siguiente ventana que aparece solicita que se le asigne el nombre que va a tener el recopilador y el modo en el que se va a crear.

En este caso se pulsara la pestaña **Crear manualmente (avanzado)** tal y como se ve en la Figura 5.2.

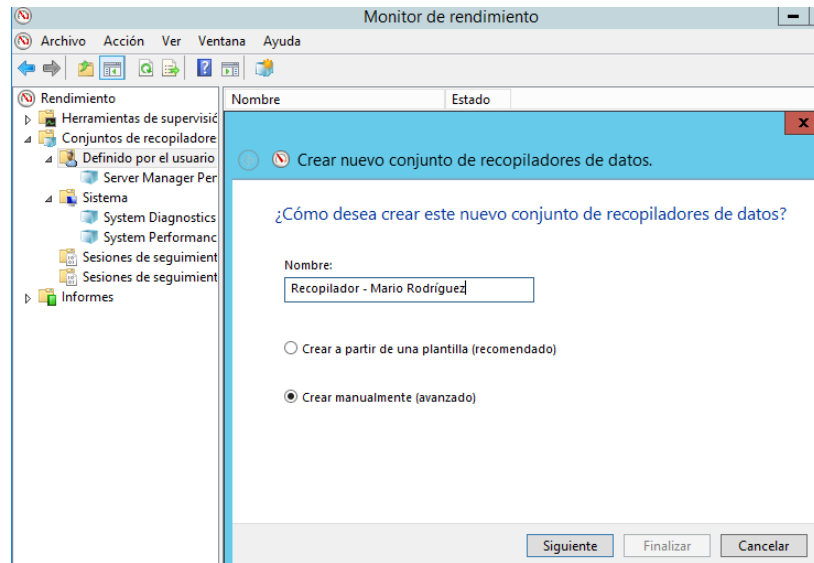


Figura 5.2: Nombre y modo de creación del conjunto

Ahora hay que elegir qué tipo de datos se desea incluir. En el enunciado de este ejercicio se piden los siguientes:

- ✓ Contador de rendimiento
- ✓ Datos de seguimiento

Se pulsan las pestañas mencionadas anteriormente (como enseña la Figura 5.3) y se hace clic en siguiente.

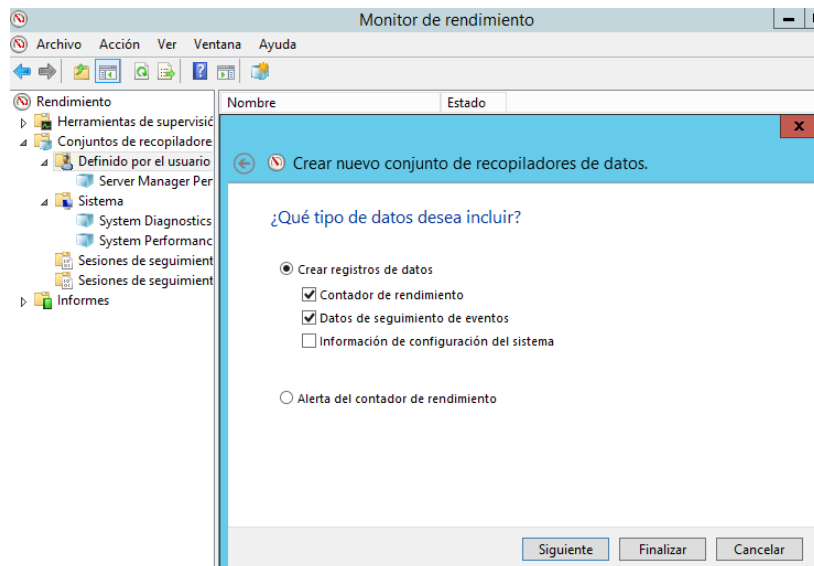


Figura 5.3: Tipos de datos a incluir en el recopilador

A continuación aparecerá una nueva ventana en la que habrá que especificar qué contadores de rendimientos se desea registrar. Para ello habrá que pulsar sobre **Agregar** (marcado con una flecha roja en la Figura 5.4)

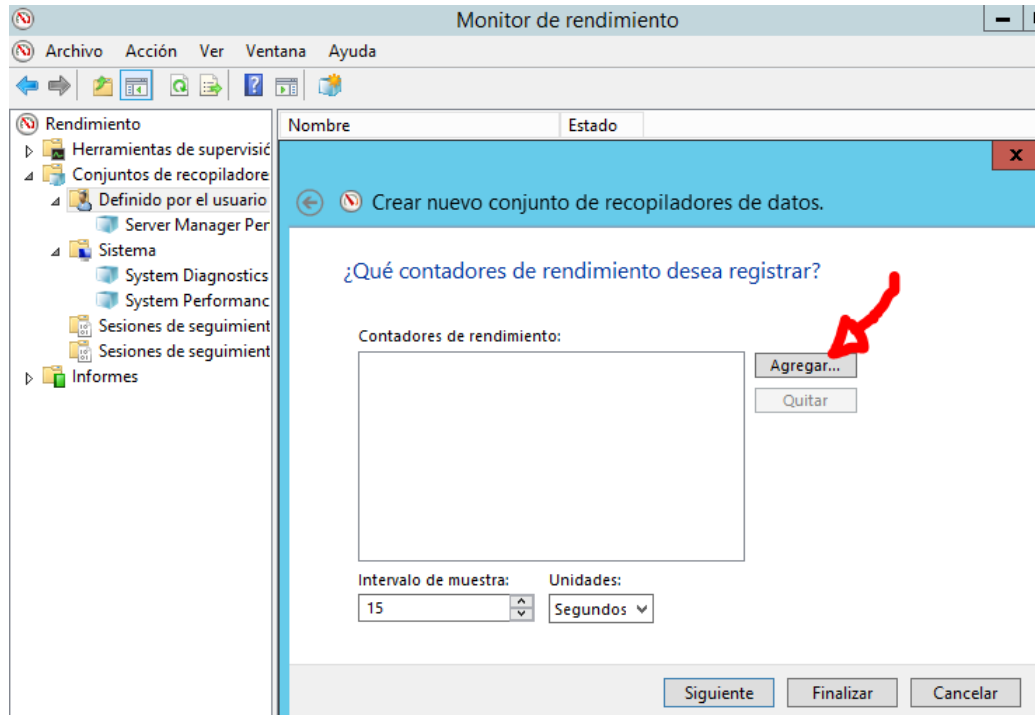


Figura 5.4: Qué contadores de rendimiento registrar

Después de pulsar **Agregar** habrá que incluir los contenedores que se deseen en la lista. En la parte izquierda de la ventana aparece un panel con todos los contadores disponibles (en este caso en el **Equipo local**). Se buscarán los tres que se piden en el enunciado:

- ✓ Procesador
- ✓ Proceso
- ✓ Servicio web

Cada vez que se seleccione un contenedor, hay que elegir también las instancias del objeto (<**Todas las instancias**>) y, de forma seguida, habrá que pulsar sobre el botón **Agregar** que se encontrará más abajo de la misma ventana.

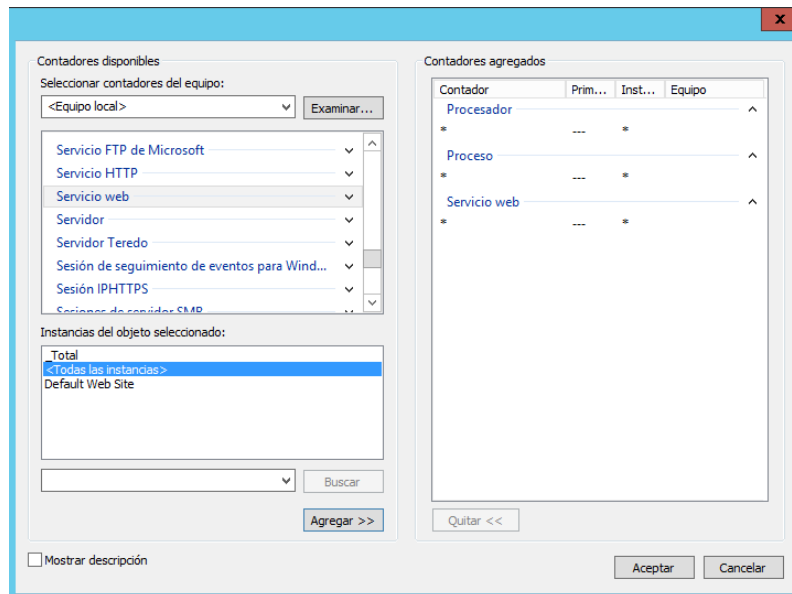


Figura 5.5: Agregar contenedores

Una vez agregados dichos contenedores deben de aparecer en el panel de la derecha de la ventana **Contenedores agregados**, tal y como muestra la Figura 5.5.

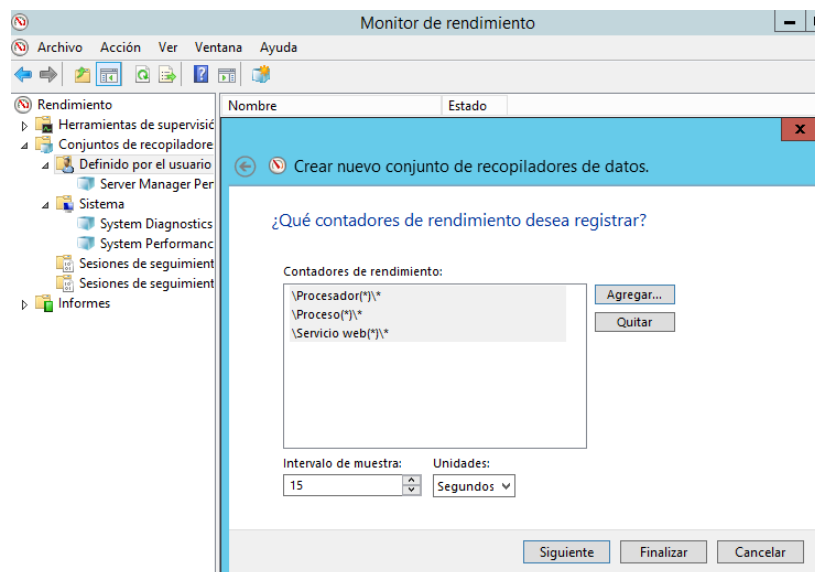


Figura 5.6: Contadores de rendimiento para registrar

Cerrada la ventana anterior, vuelve a aparecer la misma en la que se encontraba el proceso pero, esta vez, con los contadores de rendimiento incluidos en el registro (Figura 5.6)

En ésta solo queda asegurar que el intervalo de muestra sea de 15 segundos que puede verse en la Figura 5.6, tal y como pide el enunciado del ejercicio.

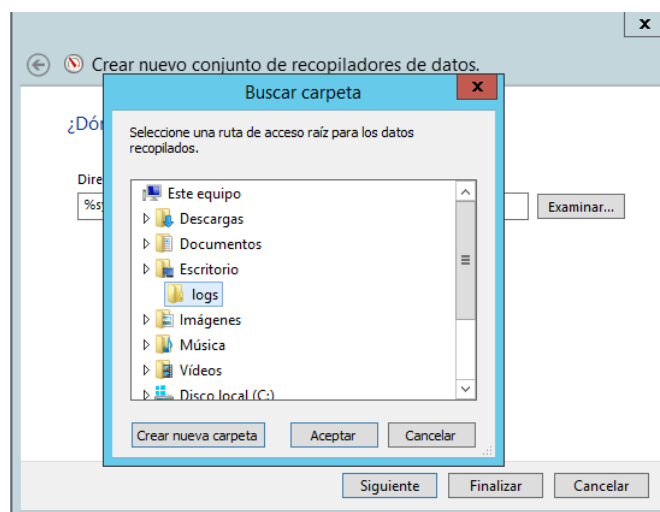


Figura 5.7: Directorio de destino del informe

Para finalizar el proceso de creación del conjunto de recopiladores solo queda especificar el directorio donde se van a almacenar los resultados, en el que en este caso sera **Escritorio\logs**. Si la carpeta no existe, se creará como aparece en la Figura 5.7.

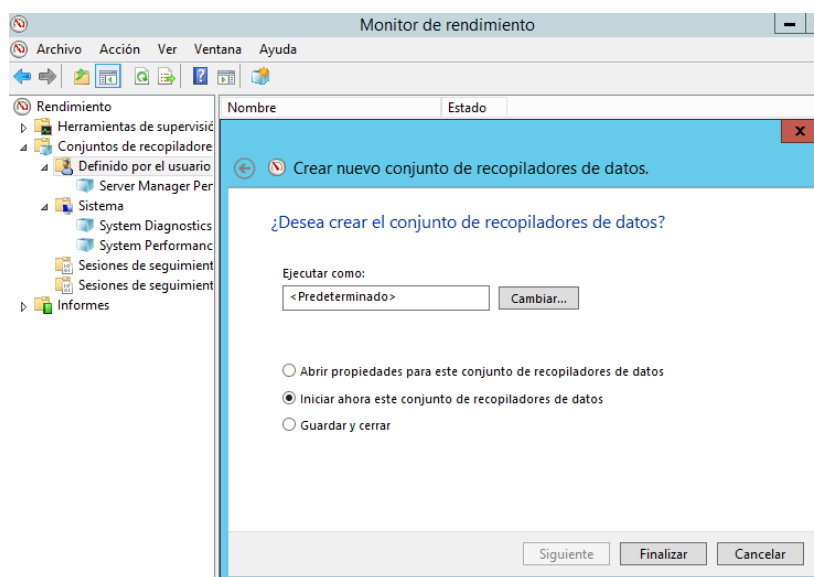


Figura 5.8: Finalizado del proceso de creación de recopiladores

En la última ventana (como la de la Figura 5.8) que aparece hay que especificar que se desea que se inicie ahora el conjunto de recopiladores y **Finalizar**.

Ya se puede comprobar cómo se encuentra **activo** el recopilador recientemente creado. En la Figura 5.9 puede verse que el funcionamiento parece correcto.

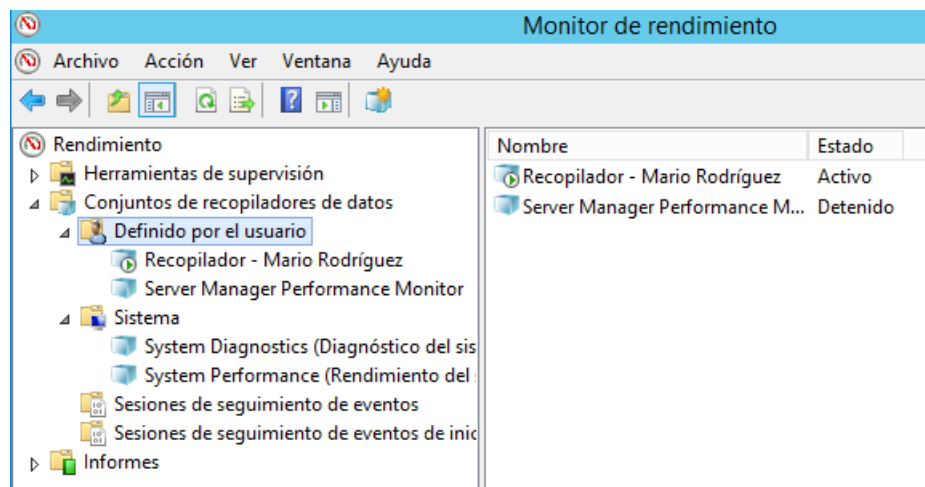


Figura 5.9: Recopilador definido activo

Para poder visualizar los resultados hay que detener el recopilador y pulsar sobre el informe que se ha creado. Se tienen dos opciones:

1. Desde el directorio que se le ha especificado en las propiedades (Figura 5.10)

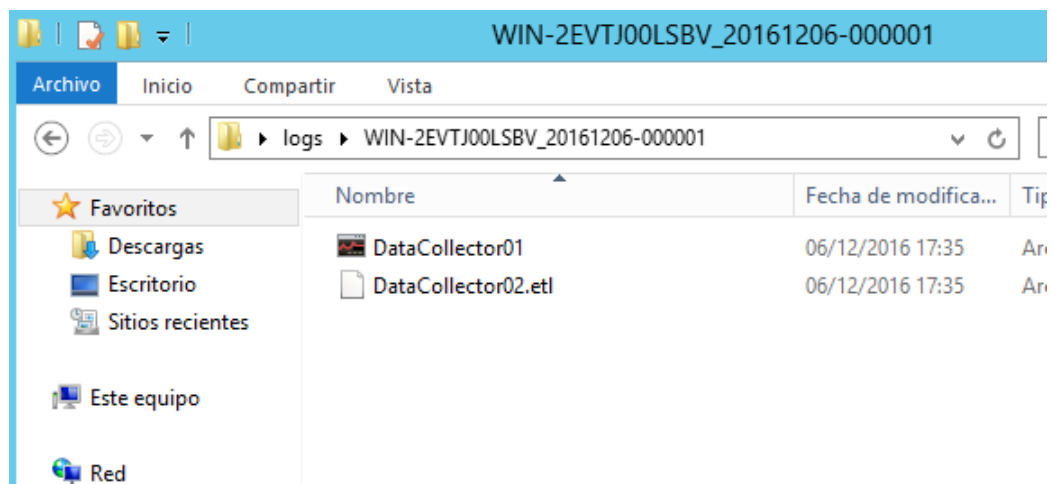


Figura 5.10: Directorio de almacenamiento de resultados

2. Desde el Monitor de rendimiento, Figura 5.11, **Panel izquierdo Rendimiento**
→ **Informes** → **Definido por el usuario** → **Nombre del recopilador**

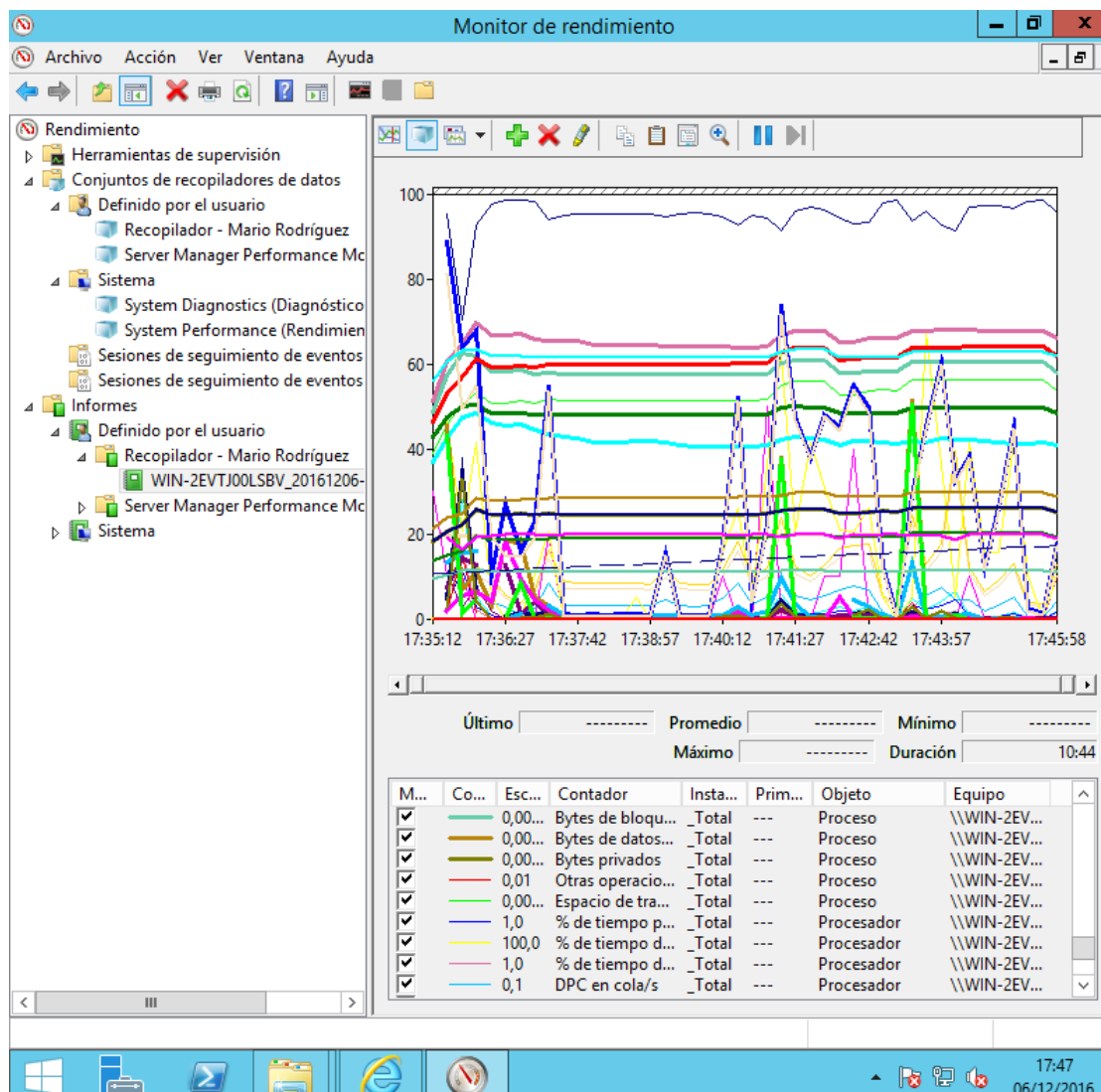


Figura 5.11: Informe del recopilador en Monitor de rendimiento

6. Cuestión 6

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

La demo que proporciona Munin permite ver una gran cantidad de gráficas relativas a distintas categorías. En este caso se han elegido las necesarias para mostrar su funcionamiento.

Cada monitorización muestra cuatro gráficas: diaria, semanal, mensual y anual. Además, también aparecen valores actuales, mínimos, medios y máximos. En la sección de disco se puede ver las entradas y salidas por dispositivo que, en el caso de este servidor, contiene solo un disco así que solo muestra una línea.

En la Figura 6.1 aparece el número de operaciones IO por segundo (en azul) y el tamaño medio de estas solicitudes (en amarillo). Los datos son de los días lunes, martes y jueves.



Figura 6.1: IOs for /dev/sda [9]

La Figura 6.2 muestra memoria usada por el sistema de ayer a hoy. En la gráfica aparecen los resultados obtenidos a través del uso de aplicaciones, buffers, caché, tablas de paginación, uso de swap...

Aparecen cuatro columnas con la ocupación de cada tipo de uso divididas por un mínimo y un máximo, así como la media entre ellas.

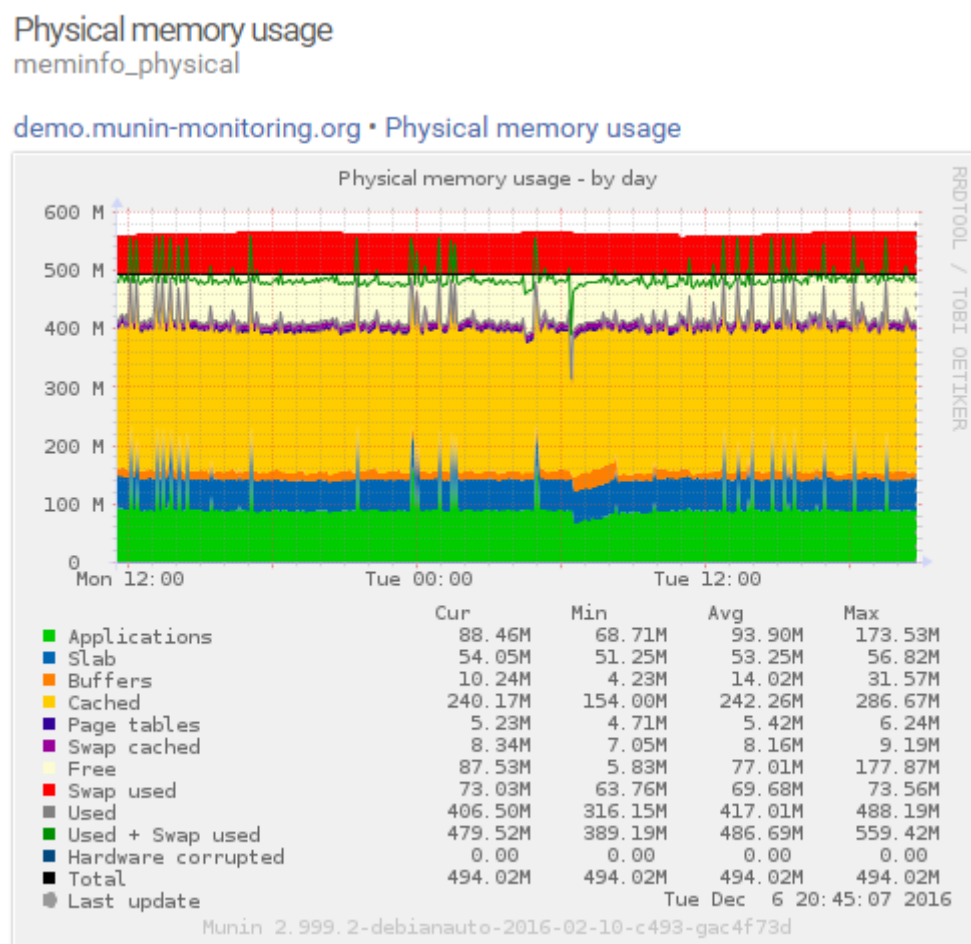


Figura 6.2: Uso de memoria física [9]

La siguiente Figura (Figura 6.3) presenta los resultados que se han obtenido por la carga media del sistema desde ayer a hoy.

Aparecen tres medidas distintas en la gráfica: de cada minuto, de cada 5 minutos y de cada 15 minutos, clasificadas también en las cuatro columnas anteriormente mencionadas.

Que se hagan esas tres medidas diferentes es indispensable para que pueda apreciarse resultados claros en la carga diaria. Basta con comparar los resultados obtenidos cada 1

min con los de 15 minutos: en los de cada 15 minutos los cambios en la carga son inapreciables, mientras que en el contrario cada medida dista en gran cantidad con respecto a la anterior.

Load average

load • system

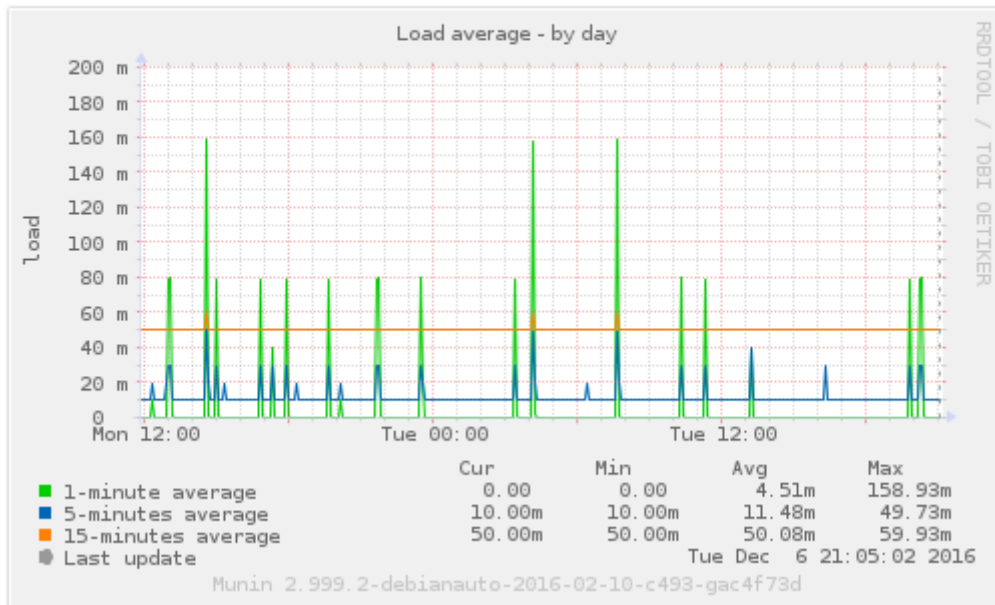


Figura 6.3: Load average [10]

La gráfica siguiente, la de la Figura 6.4, muestra el número para cada tipo de proceso que se ha ejecutado en el sistema de ayer hasta hoy.

La mayor parte de los procesos han estado dormidos casi todo el tiempo en el que se han tomado los resultados. Hay una mínima parte en el que sí se ha estado activo algún proceso de forma ininterrumpida.

Processes

processes • processes

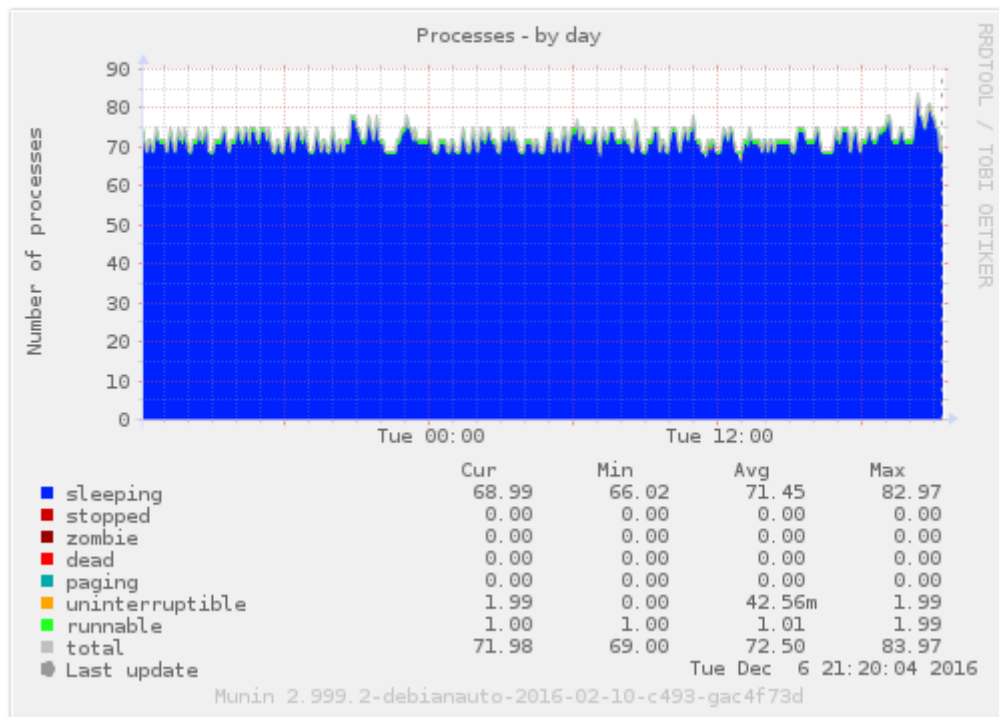


Figura 6.4: Procesos [11]

7. Cuestión 7

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

Lee Latham en el blog web **Softlayer** [5], explica el funcionamiento bastante detallado de **strace**. Según él, **strace** sirve para observar las llamadas al sistema que hace un ejecutable.

Se trata de una herramienta que puede ayudar ante una situación en la que un ejecutable termina su ejecución sin saber por qué y, a pesar de ello, no aparece rastros de esta ejecución fallida en ningún registro o, por el contrario, no es útil la información que proporciona.

En el blog, además, se hace un ejemplo de utilización de **strace**:

Primero el usuario se identifica como **root** y crea un nuevo archivo llamado **test.txt**. A continuación, se ejecuta **strace cat test.txt** y el programa retorna todas las llamadas al sistema. Además, se realiza un **open** de **test.txt** y un **read** del contenido, entre otras cosas.

Después, se realiza un ejemplo real mediante **apache** para conocer el paredero en el que se van a escribir sus logs. Para ello, lo que se hace es ejecutar **strace** con el reinicio del servicio **httpd**, escogiendo sólo las llamadas al sistema de **open**.

En esta ocasión lo que se muestra son las llamadas que se producen para abrir los archivos de registro al iniciar el servicio y, posteriormente, conocer dónde se guardan los registros de dicho servicio.

8. Cuestión 8

8.1. Escriba un script en Python o PHP

```
1  #!/usr/bin/python
2
3  import os
4  import sys
5  import cProfile
6  import re
7  cProfile.run('re.compile("foo|bar")')
8
9  nuevo = sys.argv[1]
10
11 os.chdir("/etc/ssh/")
12 f = open("sshd_config", 'r')
13 chain = f.read()
14
15 print "Cambiando el puerto de ssh..."
16 chain = chain.replace("Port 22", "Port " + nuevo)
17 f.close()
18
19 otro = open("/etc/ssh/sshd_config", 'w')
20 otro.write(chain)
21 print "Puerto ssh modificado. Su nuevo valor es ", nuevo
22
23 print "Reinicio del servicio ssh para efectuar cambios..."
24 os.system('service sshd restart')
25 otro.close()
```

fuentes/sshPython.py

El script lo que hace es cambiar la línea del fichero de configuración de ssh (`/etc/ssh/sshd_config`) en la que se especifica el puerto por el que funciona el servicio.

El puerto nuevo será un número pasado como argumento y, por supuesto, la ejecución del script debe realizarse con **permisos root** para su correcta ejecución.

En la Figura 8.1 se muestra un ejemplo de su ejecución en una máquina de Ubuntu. Primero se comprueba cuál es el puerto actual, a continuación se ejecuta el script y por último se comprueba de nuevo el fichero de configuración para ver si se ha cambiado correctamente.


```

MRR mié dic 07> cat /etc/ssh/sshd_config | grep Port
Port 22
MRR mié dic 07> sudo python sshPython.py 8521
Cambiando el puerto de ssh...
Puerto ssh modificado. Su nuevo valor es 8521
Reinicio del servicio ssh para efectuar cambios...
Rather than invoking init scripts through /etc/init.d, use the service(8)
utility, e.g. service ssh restart

Since the script you are attempting to invoke has been converted to an
Upstart job, you may also use the stop(8) and then start(8) utilities,
e.g. stop ssh ; start ssh. The restart(8) utility is also available.
ssh stop/waiting
ssh start/running, process 9953
MRR mié dic 07> cat /etc/ssh/sshd_config | grep Port
Port 8521
MRR mié dic 07>

```

Figura 8.1: Prueba de ejecución de script en Python

8.2. Analice el comportamiento del script usando el profiler presentado

Se va a utilizar el profiler **cProfile** [1], que es el recomendado para la mayoría de usuarios. Para poder invocarlo hay que realizar una de estas dos acciones:

1. Añadir al script de python las siguientes líneas:

```

1 import cProfile
2 import re
3 cProfile.run('re.compile("foo|bar")')

```

2. Ejecutar la siguiente orden desde la línea de comandos:

```

1 $ python -m cprofile sshPython.py

```

En este caso se ha utilizado, por comodidad, la primera opción.

Ahora para su funcionamiento, basta con realizar la ejecución del script y ya de forma automática lo hace también el profiler.

En la Figura 8.2 aparece la ejecución del script junto con el profiler y el desglose de los resultados que se han obtenido. En este caso, dado el poco conocimiento que tengo del tema, los resultados son bastante desconocidos. Es por ello que buscando información sobre cómo interpretar estos resultados encontré una herramienta que hace dicha interpretación algo más "entendible" por medio de representaciones gráficas, por ejemplo.

```

MRR mié dic 07> sudo python sshPython.py 29
194 function calls (189 primitive calls) in 0.000 seconds

Ordered by: standard name

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
1      0.000    0.000    0.000    0.000 <string>:1(<module>)
1      0.000    0.000    0.000    0.000 re.py:188(compile)
1      0.000    0.000    0.000    0.000 re.py:226(_compile)
1      0.000    0.000    0.000    0.000 sre_compile.py:178(_compile_charset)
1      0.000    0.000    0.000    0.000 sre_compile.py:207(_optimize_charset)
4      0.000    0.000    0.000    0.000 sre_compile.py:24(_identityfunction)
3/1    0.000    0.000    0.000    0.000 sre_compile.py:32(_compile)
1      0.000    0.000    0.000    0.000 sre_compile.py:361(_compile_info)
2      0.000    0.000    0.000    0.000 sre_compile.py:474(isstring)
1      0.000    0.000    0.000    0.000 sre_compile.py:480(_code)
1      0.000    0.000    0.000    0.000 sre_compile.py:495(compile)
5      0.000    0.000    0.000    0.000 sre_parse.py:126(__len__)
12     0.000    0.000    0.000    0.000 sre_parse.py:130(__getitem__)
7      0.000    0.000    0.000    0.000 sre_parse.py:138(append)
3/1    0.000    0.000    0.000    0.000 sre_parse.py:140(getwidth)
1      0.000    0.000    0.000    0.000 sre_parse.py:178(__init__)
10     0.000    0.000    0.000    0.000 sre_parse.py:182(__next)
2      0.000    0.000    0.000    0.000 sre_parse.py:195(match)
8      0.000    0.000    0.000    0.000 sre_parse.py:201(get)
1      0.000    0.000    0.000    0.000 sre_parse.py:301(_parse_sub)
2      0.000    0.000    0.000    0.000 sre_parse.py:379(_parse)
1      0.000    0.000    0.000    0.000 sre_parse.py:663(parse)
1      0.000    0.000    0.000    0.000 sre_parse.py:67(__init__)
3      0.000    0.000    0.000    0.000 sre_parse.py:90(__init__)
1      0.000    0.000    0.000    0.000 {_sre.compile}
15     0.000    0.000    0.000    0.000 {isinstance}
38/37  0.000    0.000    0.000    0.000 {len}
2      0.000    0.000    0.000    0.000 {max}
48     0.000    0.000    0.000    0.000 {method 'append' of 'list' objects}
1      0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
1      0.000    0.000    0.000    0.000 {method 'get' of 'dict' objects}
1      0.000    0.000    0.000    0.000 {method 'items' of 'dict' objects}
8      0.000    0.000    0.000    0.000 {min}
6      0.000    0.000    0.000    0.000 {ord}

```

Figura 8.2: Primera aproximación de ejecución del profiler

Esta herramienta se llama **KCacheGrind** [12] en la que se encuentra la opción **pyprof2calltree** que lo que hace es convertir un archivo con un volcado de datos del profiler en uno compatible para si mismo, para su posterior representación en la aplicación.

Por tanto, se instaló la herramienta por medio de la orden:

```
1 $ sudo apt-get install kcacheGrind
```

Hubo un problema a la hora de ejecutar **pyprof2calltree**, ya que ésta no se encontraba dentro de la herramienta y no encontré un comando para instalarla mediante **apt**. Busqué el paquete y lo descargué desde su página (<https://pypi.python.org/pypi/pyprof2calltree/>).

Lo que se descarga es un fichero en python, por lo que para ejecutarla se debe de hacer como un script cualquiera de python con la peculiaridad de que debe de tener unos argumentos específicos, que son:

```
1 $ pyprof2calltree.py [-k] [-o output\_file\_path] [-i input\_file\_path] [-r scriptfile [args]]
```

Una vez instalada la herramienta con todos sus componentes necesarios ya solo falta la parte más importante: el archivo con el volcado de datos. Para ello basta con completar la línea donde se escribe la función run del cProfile con el nombre del archivo de salida. En este caso, la línea 7 del script que antes de la modificación era:

```
1 cProfile.run('re.compile("foo|bar")')
```

Y, posteriormente a la modificación:

```
1 cProfile.run('re.compile("foo|bar")', 'salida.cprof')
```

A continuación se procede a la conversión del fichero con el volcado de los resultados (que previamente se han obtenido al ejecutar el script personal), para ello se ejecuta la siguiente orden, tal y como muestra la Figura 8.3:

```
1 python pyprof2calltree.py -k -i salida.cprof
```

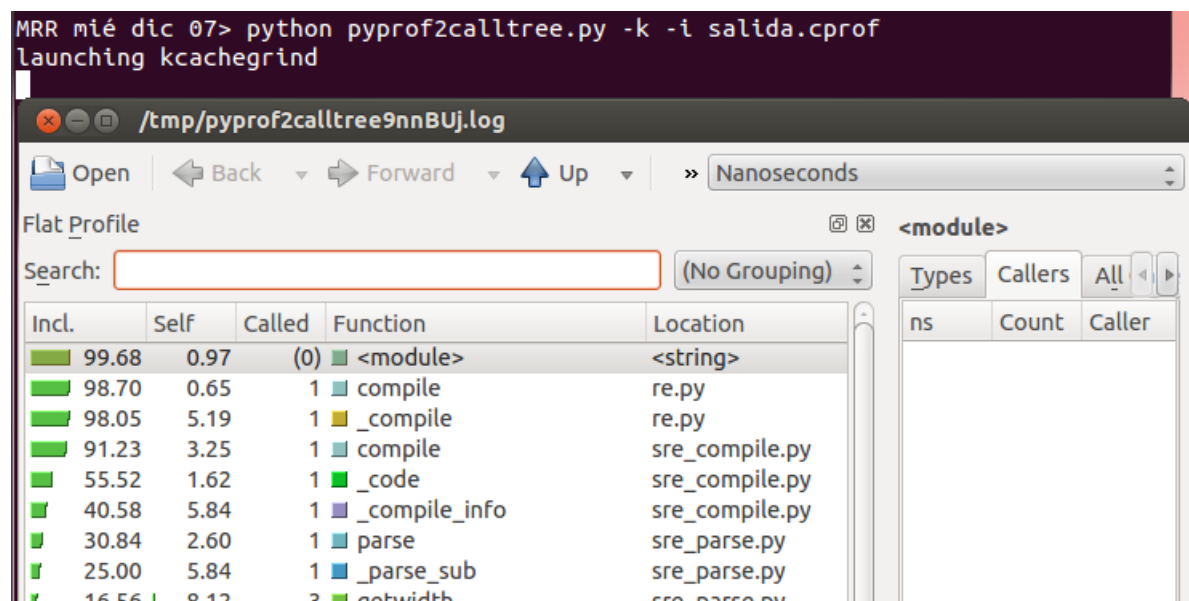


Figura 8.3: Presentación de los datos obtenidos con Kcachegrind

Finalmente, ya sí que se han obtenido un volcado de los resultados del profiler mejor representados. Basta con observar la Figura 8.4.

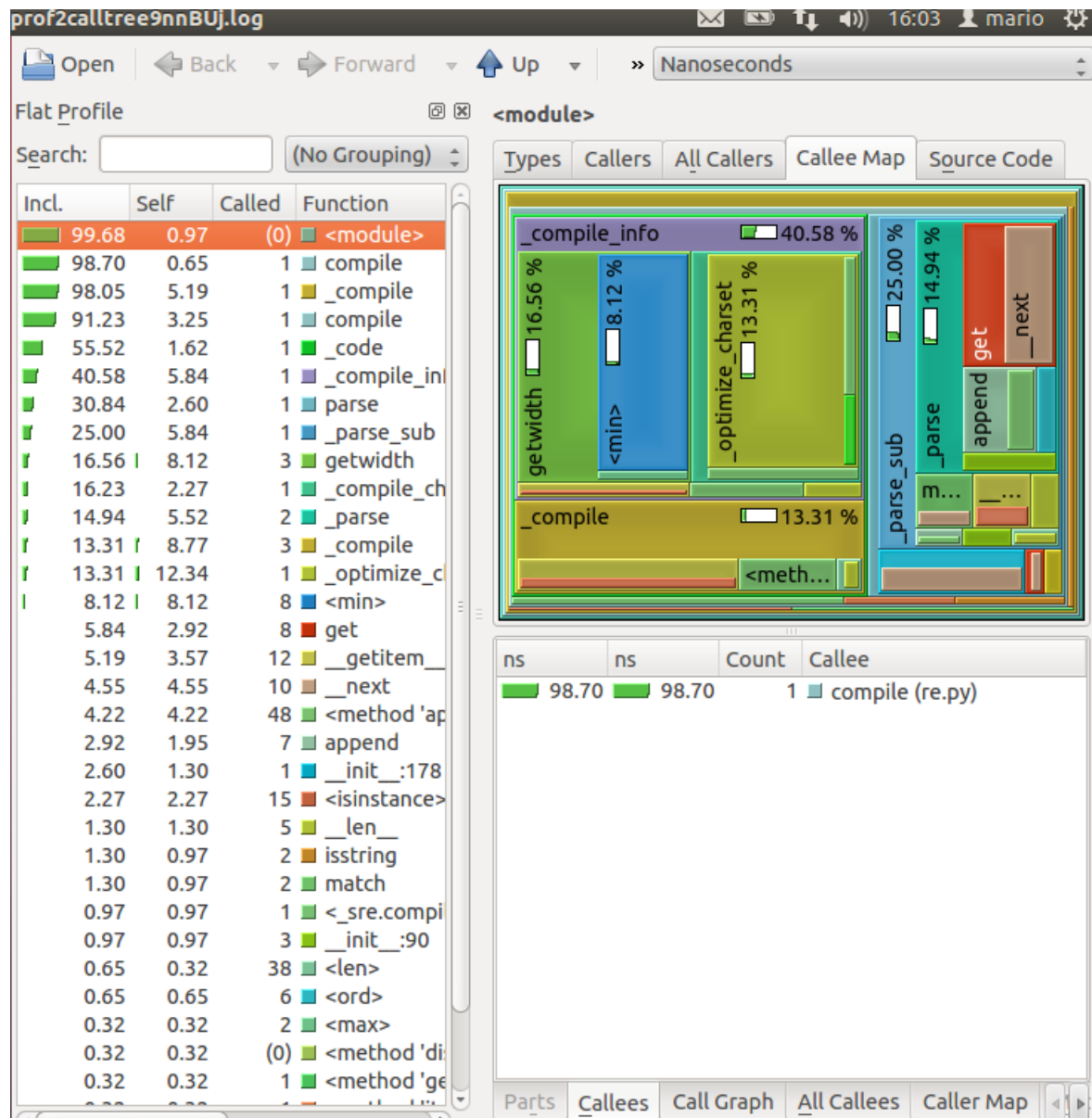


Figura 8.4: Primera aproximación de ejecución del profiler

La Figura 8.5 aparece una gráfica en forma de árbol obtenida también en Kcachegrind (desde **Call graph**) en el que se presenta el tiempo que gastan las llamadas al sistema en una ejecución concreta, así como la relación coexistente entre cada llamada.

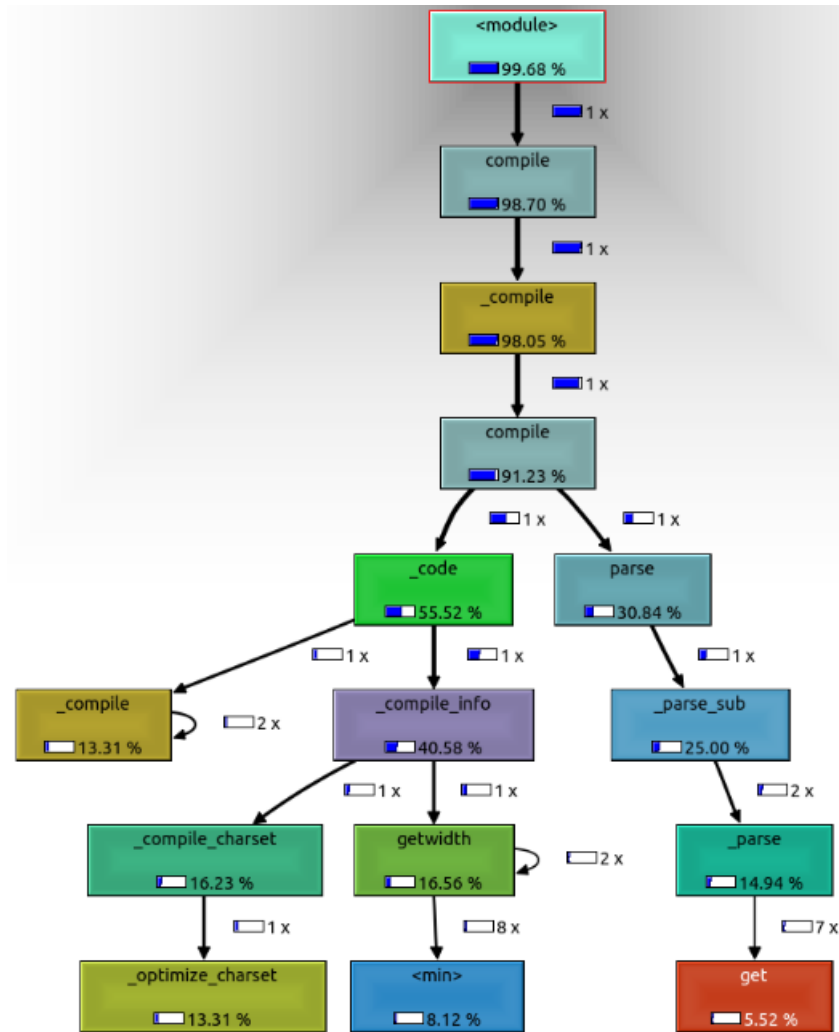


Figura 8.5: Representación de la organización del tiempo de las llamadas

En ésta se puede ver cómo la función **module** (llamada 0) y a **compile** son la que más tiempo requieren en este script mientras que, por el contrario, la llamada **get** (llamada 8) es la que menos.

9. Cuestión 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)

En este caso se ha realizado el ejercicio a través de la **consola mysql**.

En primer lugar, una vez en la consola MariaDB o Mysql, se crea la base de datos para poder hacer la prueba mediante la orden **'CREATE DATABASE nombre_database ;'** (sin las comillas).

A continuación, con la orden **'USE nombre_database ;'** se accede a la database creada para poder trabajar sobre ella.

Todo este proceso anterior puede verse en la Figura 9.1.

```
MRR mié dic 07> mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 5.5.50-MariaDB MariaDB Server

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MRR Wed Dec  7 19:38:27 2016 [(none)]> CREATE DATABASE dataMario;
Query OK, 1 row affected (0.00 sec)

MRR Wed Dec  7 19:38:37 2016 [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| dataMario |
| mysql |
| performance_schema |
+-----+
4 rows in set (0.00 sec)

MRR Wed Dec  7 19:38:56 2016 [(none)]> USE dataMario;
Database changed
MRR Wed Dec  7 19:39:11 2016 [dataMario]> █
```

Figura 9.1: Creación de una nueva base de datos en Mysql

Ahora, ya se pueden crear tablas sobre la base de datos creada. Se crearán dos como ejemplo: una llamada pueblo y otra llamada país a través de la orden **CREATE TABLE nombre_tabla (nombre_column tipo_column)** ; mostrándolas seguidamente con **SHOW TABLES** ; como aparece en la Figura 9.2.

```

MRR Wed Dec 7 19:41:04 2016 [dataMario]>
MRR Wed Dec 7 19:41:04 2016 [dataMario]> CREATE TABLE pueblo (nombre CHAR(20) NOT NULL, provincia CHAR(20) NULL);
Query OK, 0 rows affected (0.01 sec)

MRR Wed Dec 7 19:41:53 2016 [dataMario]> CREATE TABLE pais (nombre CHAR(20) NOT NULL, continente CHAR(20) NULL);
Query OK, 0 rows affected (0.04 sec)

MRR Wed Dec 7 19:42:02 2016 [dataMario]> SHOW TABLES;
+-----+
| Tables_in_dataMario |
+-----+
| pais                 |
| pueblo               |
+-----+
2 rows in set (0.00 sec)

MRR Wed Dec 7 19:42:10 2016 [dataMario]> █

```

Figura 9.2: Creación de tablas en Mysql

Una vez las tablas están creadas, hay que comprobar si el profiler se encuentra en funcionamiento. Para ello basta con ejecutar la orden **SELECT @@profiling ;** y ver si su valor es **uno** (si es **cero** es que no está activo).

Como se muestra en la Figura 9.3 tras la primera comprobación su valor es **cero**. Para cambiarlo se utiliza la orden **SET profiling=1 ;**, como aparece en esta misma figura.

```

MRR Wed Dec 7 19:44:14 2016 [dataMario]> SELECT @@profiling;
+-----+
| @@profiling |
+-----+
| 0           |
+-----+
1 row in set (0.00 sec)

MRR Wed Dec 7 19:44:25 2016 [dataMario]> SET profiling=1;
Query OK, 0 rows affected (0.00 sec)

MRR Wed Dec 7 19:44:33 2016 [dataMario]> SELECT @@profiling;
+-----+
| @@profiling |
+-----+
| 1           |
+-----+
1 row in set (0.00 sec)

MRR Wed Dec 7 19:44:43 2016 [dataMario]> █

```

Figura 9.3: Activación del profiler en Mysql

Para poder obtener resultados con el profiler hay que realizar alguna consulta [7]. En la Figura 9.4 presenta la consulta que se ha hecho en este caso (con la orden **SELECT** para seleccionar datos de una tabla) y los resultados que se han obtenido después con el profiler con la orden **SHOW PROFILES** ; con la que se muestra información detallada acerca de **un solo estado**, la declaración ejecutada más recientemente.

```
MRR Wed Dec 7 19:45:39 2016 [dataMario]>
MRR Wed Dec 7 19:45:39 2016 [dataMario]> SELECT nombre,provincia from puebl
o;
Empty set (0.00 sec)

MRR Wed Dec 7 19:45:59 2016 [dataMario]> SHOW PROFILES;
+-----+-----+-----+
| Query_ID | Duration | Query |
+-----+-----+-----+
| 1 | 0.00009219 | SELECT @@profiling |
| 2 | 0.00030546 | SELECT nombre,provincia from pueblo |
+-----+-----+-----+
2 rows in set (0.00 sec)

MRR Wed Dec 7 19:46:10 2016 [dataMario]> █
```

Figura 9.4: Utilización profiler en Mysql

Por otro lado, hay otra orden en la que en cambio aparece una **lista** de las declaraciones más recientes enviadas al servidor. Ésta es **SHOW PROFILE** ; [7] y puede verse un ejemplo de uso en la Figura 9.5.

```
MRR Wed Dec 7 19:48:22 2016 [dataMario]> SHOW PROFILE;
+-----+-----+
| Status | Duration |
+-----+-----+
| starting | 0.000047 |
| checking permissions | 0.000007 |
| Opening tables | 0.000106 |
| After opening tables | 0.000006 |
| System lock | 0.000004 |
| Table lock | 0.000002 |
| After table lock | 0.000004 |
| init | 0.000014 |
| optimizing | 0.000007 |
| statistics | 0.000012 |
| preparing | 0.000008 |
| executing | 0.000003 |
| Sending data | 0.000056 |
| end | 0.000004 |
| query end | 0.000004 |
| closing tables | 0.000006 |
| freeing items | 0.000004 |
| updating status | 0.000008 |
| cleaning up | 0.000002 |
+-----+-----+
19 rows in set (0.00 sec)
```

Figura 9.5: Utilización profiler en Mysql

Para especificar sobre qué consulta se quiere ver el registro, hay que añadir a **SHOW PROFILE** el operador **FOR QUERY Query_ID** (ver Figura 9.6), donde **Query_ID** es el ID de consulta tal y como se ha listado en **SHOW PROFILES** ;

```
MRR Thu Dec 8 20:01:48 2016 [dataMario]> SHOW PROFILES;
+-----+-----+-----+
| Query_ID | Duration | Query |
+-----+-----+-----+
| 1 | 0.00009110 | SELECT @@profiling |
| 2 | 0.00025515 | SELECT nombre,provincia from pueblo |
+-----+-----+-----+
2 rows in set (0.00 sec)

MRR Thu Dec 8 20:01:49 2016 [dataMario]> SHOW PROFILE MEMORY FOR QUERY 2 ;
+-----+-----+
| Status | Duration |
+-----+-----+
| starting | 0.000039 |
| checking permissions | 0.000008 |
| Opening tables | 0.000014 |
| After opening tables | 0.000005 |
| System lock | 0.000004 |
| Table lock | 0.000003 |
| After table lock | 0.000004 |
| init | 0.000014 |
| optimizing | 0.000007 |
| statistics | 0.000015 |
| preparing | 0.000009 |
| executing | 0.000003 |
| Sending data | 0.000099 |
| end | 0.000006 |
| query end | 0.000005 |
| closing tables | 0.000007 |
| freeing items | 0.000004 |
| updating status | 0.000008 |
| cleaning up | 0.000003 |
+-----+-----+
19 rows in set (0.00 sec)
```

Figura 9.6: Utilización profiler en Mysql

Sin embargo, si lo que se quiere es que se muestre un tipo concreto de registro se le puede especificar justo antes del operador **FOR**, como en el caso de la Figura 9.6, donde se le ha especificado que muestre los datos de uso de **memoria**.

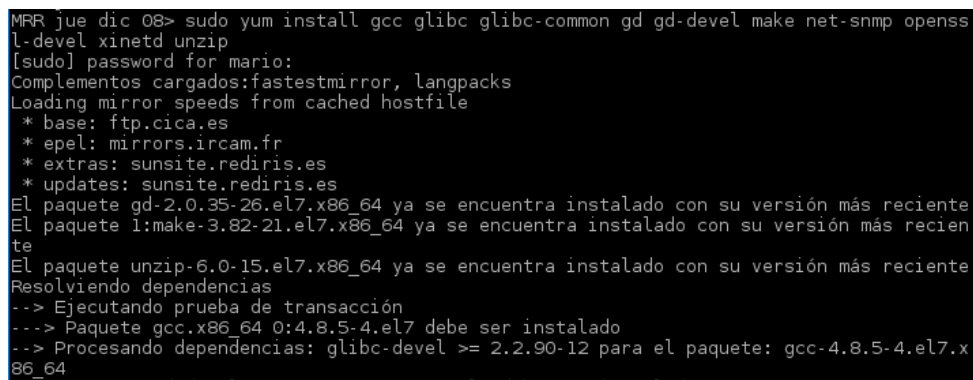
10. Cuestión opcional 2

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

Se va a proceder a instalar **Nagios Core** en Centos 7. [8]

- Instalación de los paquetes necesarios (Figura 10.1) para satisfacer las dependencias de la herramienta mediante la orden:

```
1 > sudo yum install gcc glibc glibc-common gd gd-devel  
make net-snmp openssl-devel xinetd unzip
```

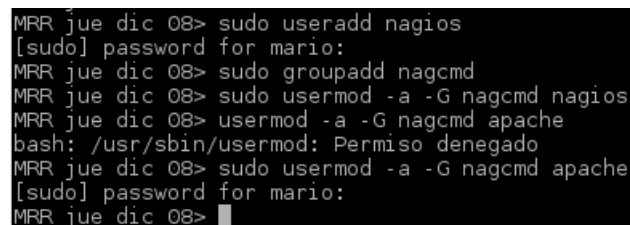


```
MRR jue dic 08> sudo yum install gcc glibc glibc-common gd gd-devel make net-snmp openss  
l-devel xinetd unzip  
[sudo] password for mario:  
Complementos cargados:fastestmirror, langpacks  
Loading mirror speeds from cached hostfile  
* base: ftp.cica.es  
* epel: mirrors.ircam.fr  
* extras: sunsite.rediris.es  
* updates: sunsite.rediris.es  
El paquete gd-2.0.35-26.el7.x86_64 ya se encuentra instalado con su versión más reciente  
El paquete 1:make-3.82-21.el7.x86_64 ya se encuentra instalado con su versión más reciente  
El paquete unzip-6.0-15.el7.x86_64 ya se encuentra instalado con su versión más reciente  
Resolviendo dependencias  
--> Ejecutando prueba de transacción  
--> Paquete gcc.x86_64 0:4.8.5-4.el7 debe ser instalado  
--> Procesando dependencias: glibc-devel >= 2.2.90-12 para el paquete: gcc-4.8.5-4.el7.x  
86_64
```

Figura 10.1: Instalación de los paquetes necesarios para Nagios

- Creación de usuario y grupo (Figura 10.2), para que se ejecute el proceso de Nagios, a través las órdenes:

```
1 > sudo useradd nagios  
2 > sudo groupadd nagcmd  
3 > sudo usermod -a -G nagcmd nagios  
4 > sudo usermod -a -G nagcmd apache
```



```
MRR jue dic 08> sudo useradd nagios  
[sudo] password for mario:  
MRR jue dic 08> sudo groupadd nagcmd  
MRR jue dic 08> sudo usermod -a -G nagcmd nagios  
MRR jue dic 08> usermod -a -G nagcmd apache  
bash: /usr/sbin/usermod: Permiso denegado  
MRR jue dic 08> sudo usermod -a -G nagcmd apache  
[sudo] password for mario:  
MRR jue dic 08>
```

Figura 10.2: Creación de usuario y grupo para Nagios

- Antes de comenzar nuestra instalación, se necesita inhabilitar **SELinux** (Figura 10.3). Para ello basta con modificar una línea en el fichero `/etc/selinux/config`:

```
1 SELINUX=disabled
```

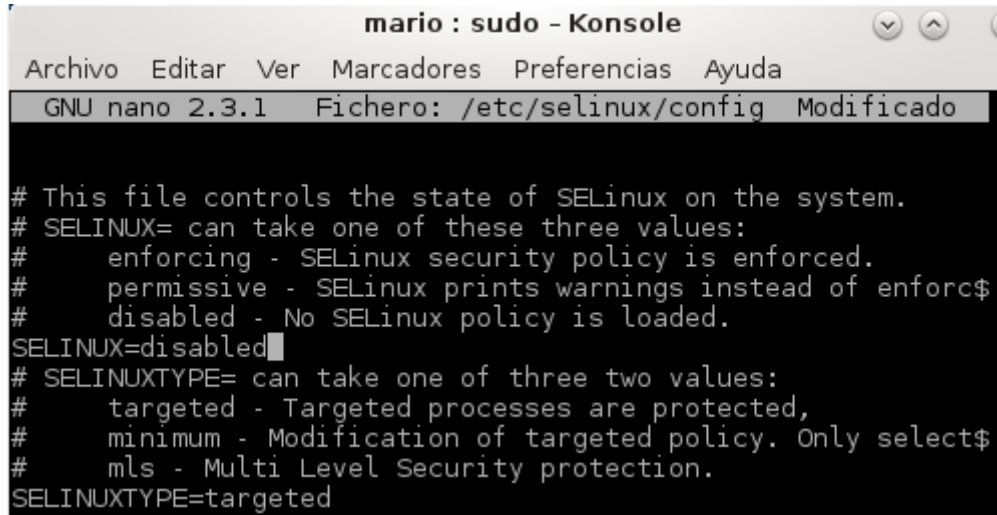


Figura 10.3: Deshabilitación de SELinux

- Descarga de Nagios y sus plugins. Con las siguientes órdenes se descargará, como aparece en la Figura 10.4, tanto la herramienta como los plugins necesarios para su funcionamiento:

```
1 > cd /tmp
2 > wget https://assets.nagios.com/downloads/nagioscore/
  releases/nagios-4.1.1.tar.gz
3 > wget http://www.nagios-plugins.org/download/nagios-
  plugins-2.1.1.tar.gz
4 > tar xzf nagios-4.1.1.tar.gz
5 > tar xzf nagios-plugins-2.1.1.tar.gz
6 > cd nagios-4.1.1
```

```

MRR jue dic 08> cd /tmp
MRR jue dic 08> wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.1.1.tar.gz
--2016-12-08 22:05:17-- https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.1.1.tar.gz
Resolviendo assets.nagios.com (assets.nagios.com)... 72.14.181.71, 2600:3c00::f03c:91ff:fedf:b821
Conectando con assets.nagios.com (assets.nagios.com)[72.14.181.71]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 11206656 (11M) [application/x-gzip]
Grabando a: "nagios-4.1.1.tar.gz"

83% [=====> ] 9.363.456 941KB/s T.E. 2s

```

Figura 10.4: Descarga de Nagios y sus plugins

- Instalación de Nagios. Ahora que todos los archivos se han descomprimido es el momento de su compilación e instalación. Tal y como aparece en las Figuras 10.5, 10.6, 10.7, 10.8; se utilizan las siguientes órdenes:

```

1 > ./configure --with-command-group=nagcmd
2 > make all
3 > make install
4 > make install-init
5 > make install-config
6 > make install-commandmode
7 > make install-webconf

```

```

MRR jue dic 08> make all
cd ./base && make
make[1]: se ingresa al directorio `/tmp/nagios-4.1.1/base'
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nagios.o nagios.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o broker.o broker.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nebmods.o nebmods.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o ../common/shared.o ../common/shared.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nerd.o nerd.c

```

Figura 10.5: Instalación de Nagios

```

MRR jue dic 08> sudo make install
cd ./base && make install
make[1]: se ingresa al directorio `/tmp/nagios-4.1.1/base'
make install-basic
make[2]: se ingresa al directorio `/tmp/nagios-4.1.1/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/
nagios/bin
/usr/bin/install -c -m 774 -o nagios -g nagios nagios /usr/lo
cal/nagios/bin
/usr/bin/install -c -m 774 -o nagios -g nagios nagiosstats /us
r/local/nagios/bin
make[2]: se sale del directorio `/tmp/nagios-4.1.1/base'
make strip-post-install

```

Figura 10.6: Instalación de Nagios

```

MRR jue dic 08> sudo make install-init
/usr/bin/install -c -m 755 -d -o root -g root /etc/rc.d/init.
d
/usr/bin/install -c -m 755 -o root -g root daemon-init /etc/r
c.d/init.d/nagios

*** Init script installed ***

MRR jue dic 08> sudo make install-config
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/
nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/
nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-conf
ig/nagios.cfg /usr/local/nagios/etc/nagios.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-conf
ig/cgi.cfg /usr/local/nagios/etc/cgi.cfg

```

Figura 10.7: Instalación de Nagios

```

MRR jue dic 08> sudo make install-commandmode
/usr/bin/install -c -m 775 -o nagios -g nagcmd -d /usr/local/
nagios/var/rw
chmod g+s /usr/local/nagios/var/rw

*** External command directory configured ***

MRR jue dic 08> sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/http
d/conf.d/nagios.conf

*** Nagios/Apache conf file installed ***

```

Figura 10.8: Instalación de Nagios

- Creación de una contraseña de acceso a Nagios. Es necesario que se cree una contraseña vinculada a la cuenta **nagiosadmin** anteriormente creada para el acceso a la herramienta. En la Figura 10.9 se ve la ejecución de la siguiente orden, en la que se requiere y se completa una contraseña.

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

```
MRR jue dic 08> sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
MRR jue dic 08> █
```

Figura 10.9: Creación de la contraseña de usuario para Nagios

- Instalación de los plugins necesarios para Nagios. Ahora que Nagios está instalado, se necesitan instalar los plugins (Figura 10.10) para que pueda utilizarlos.

```
1 > cd /tmp/nagios-plugins-2.1.1
2 > ./configure --with-nagios-user=nagios --with-nagios-group=nagios --with-openssl
3 > make all
4 > make install
```

```
MRR jue dic 08> cd /tmp/nagios-plugins-2.1.1
MRR jue dic 08> sudo ./configure --with-nagios-user=nagios --with-nagios-group=nagios --with-openssl
checking for a BSD-compatible install... /bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether to disable maintainer-specific portions of Makefiles... yes
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking for gcc... gcc
```

Figura 10.10: Instalación de los plugins necesarios para Nagios

- Apertura de los puertos para Nagios. Hay que asegurar que los puertos de conexión que utiliza la herramienta (**80/tcp**) se encuentran abiertos. Si no es así, como en este caso, hay que abrirlos mediante las órdenes (Figura 10.11) para que pueda utilizarlos.

```
1 > firewall-cmd --zone=public --add-port=80/tcp --
    permanent
2 > firewall-cmd --reload
```

```
MRR jue dic 08>
MRR jue dic 08> sudo firewall-cmd --zone=public --add-port=80
/tcp --permanent
success
MRR jue dic 08> sudo firewall-cmd --reload
success
MRR jue dic 08> █
```

Figura 10.11: Apertura de los puertos para Nagios

- Inicio de Nagios. Si todo el proceso realizado hasta ahora ha sido correcto, solo queda iniciar el servicio para su utilización, como se ve en la Figura 10.12, con las siguientes órdenes:

```
1 > service httpd start
2 > service nagios start
```

```
MRR jue dic 08> sudo service httpd start
Redirecting to /bin/systemctl start httpd.service
MRR jue dic 08> sudo service nagios start
Reloading systemd: [
OK ]
Starting nagios (via systemctl): [
OK ]
MRR jue dic 08> █
```

Figura 10.12: Inicio de Nagios

Una vez terminado el proceso de instalación, se comprobará si funciona todo correctamente. Para ello hay que acceder al navegador e ir a la dirección **http://localhost/nagios/**. A continuación se requerirá un login (Figura 10.13), que se completará con los datos que se han especificado en el proceso de instalación.

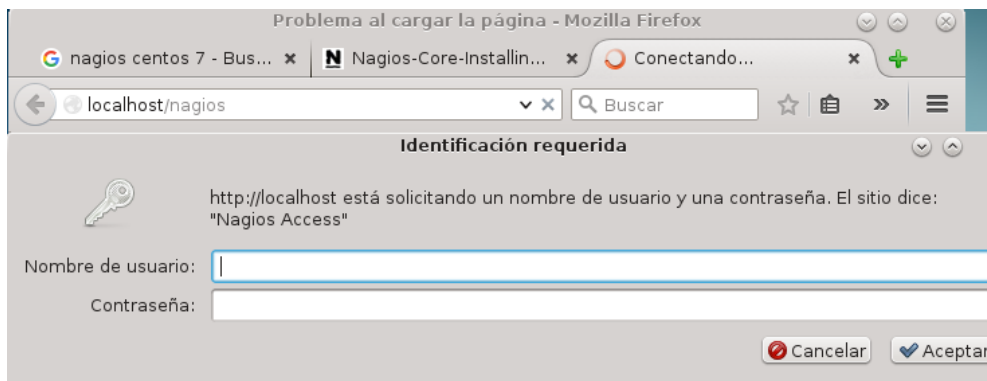


Figura 10.13: Login en Nagios

Si el login es correcto, por fin aparecerá la pantalla principal de Nagios igual que en la Figura 10.14.

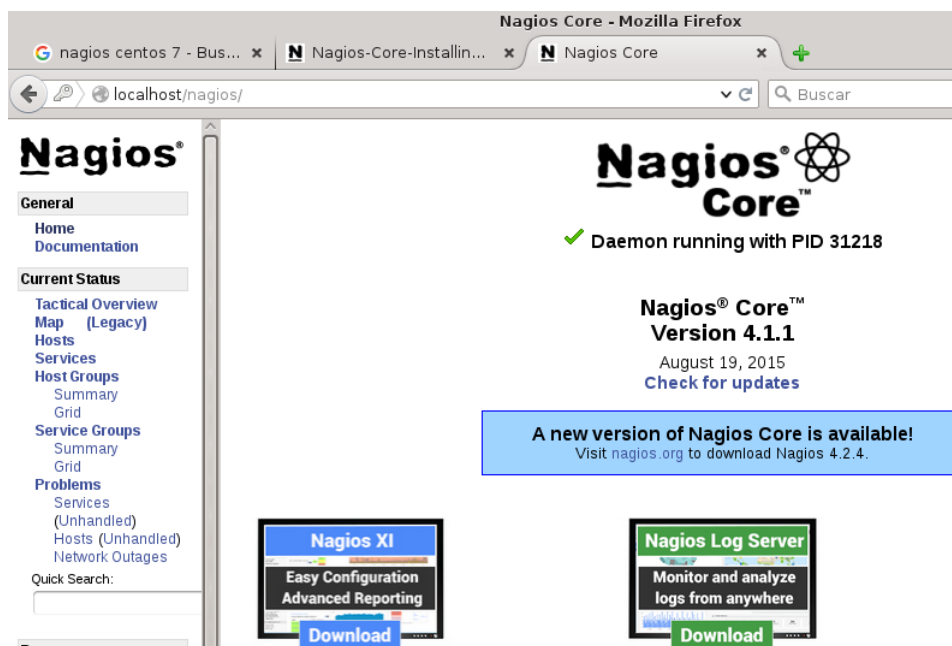


Figura 10.14: Pantalla principal de Nagios

Ahora ya se puede obtener información de monitoreo con Nagios. Si se pulsa en el panel izquierdo sobre **Tactical Overview** puede verse un resumen bastante detallado del sistema.

En éste, que es algo como la Figura 10.15, se muestra las funciones del monitoreo, el estado de los servicios, los dispositivos conectados, el rendimiento del monitoreo... etc

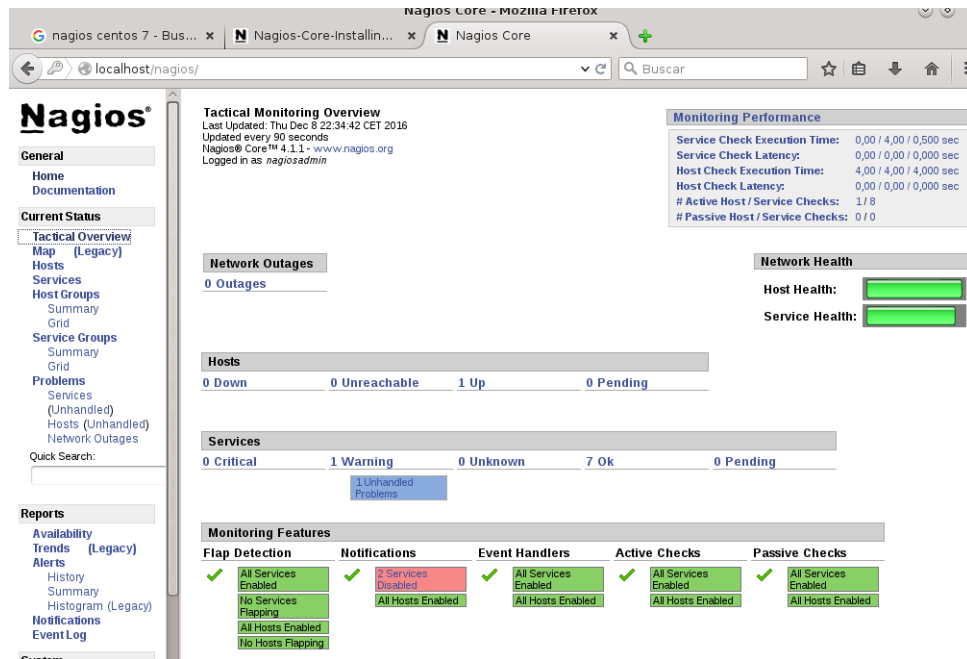


Figura 10.15: Tactical Overview Nagios

En la Figura 10.16 se muestran los detalles del estado del servicio para todos los hosts en la que, por ejemplo, muestra un aviso en el servicio HTTP ya que antes de acceder a esta herramienta he puesto mal el login y es lo que ha registrado.

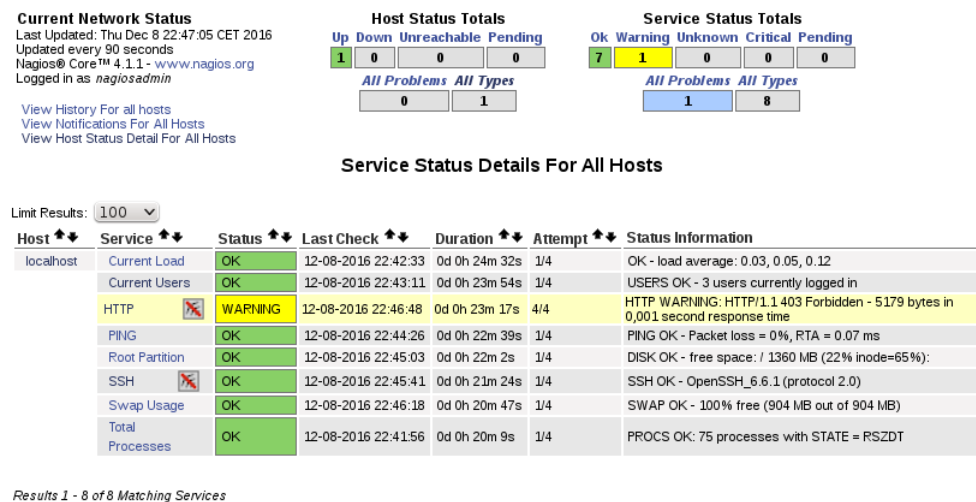


Figura 10.16: Estados de los servicios Nagios

La Figura 10.17 presenta los resultados obtenidos a través de la información del rendimiento del sistema, desde la opción **System**→**Performance info**.

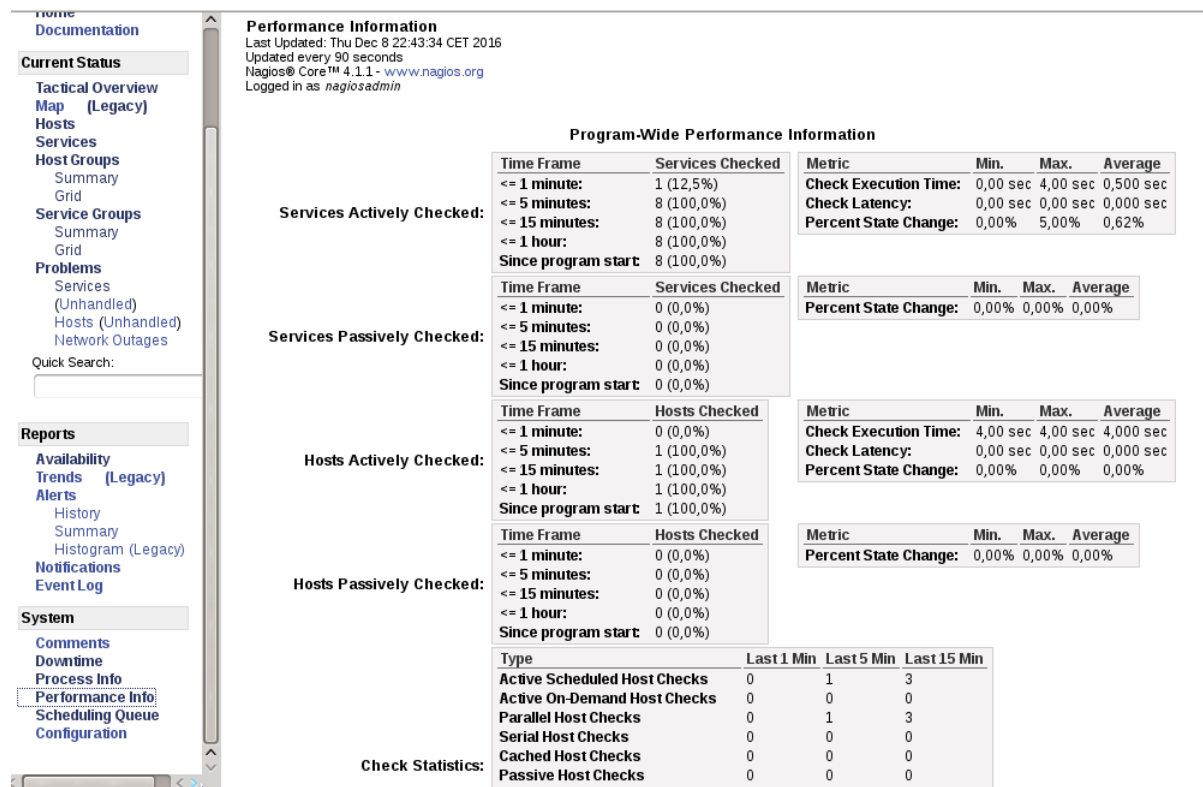


Figura 10.17: Información de rendimiento en Nagios

Éstos se muestran en tablas, clasificados en distintos periodos de tiempo para cada servicio verificado. Aparecen resultados de servicios y hosts activos/pasivos, así como las estadísticas resumidas en una tabla distinta.

Referencias

- [1] Python Docs. *The Python Profilers*. Recuperado el 4 de diciembre de 2016, desde: <http://docs.python.org/2/library/profile.html>.
- [2] CentOS Deployment Guide. *Chapter 35. Log Files*. Recuperado el 1 de diciembre de 2016, desde: https://www.centos.org/docs/5/html/Deployment_Guide-en-US/ch-logfiles.html.
- [3] CentOS Deployment Guide. *Chapter 35.1.1. Configuring Cron Tasks*. Recuperado el 1 de diciembre de 2016, desde: https://www.centos.org/docs/5/html/5.2/Deployment_Guide/s2-autotasks-cron-configuring.html.
- [4] Jered Heeschen. *Understanding logrotate utility*. Recuperado el 1 de diciembre de 2016, desde: <https://support.rackspace.com/how-to/understanding-logrotate-utility/>.
- [5] Lee Latham. *Sysadmin Tips and Tricks - Using strace to Monitor System Calls*. Recuperado el 4 de diciembre de 2016, desde: http://blog.softlayer.com/2013/sysadmin-tips-and-tricks-using-strace-to-monitor-system-calls#utm_source=twitter&utm_medium=social&utm_content=beyond-the-command-line-with-strace&utm_campaign=blog-development-tips-and-tricks.
- [6] SCSI Seagate Manual. *3.13 MODE SENSE(6) command*. Recuperado el 1 de diciembre de 2016, desde: <http://www.seagate.com/staticfiles/support/disc/manuals/scsi/100293068a.pdf>.
- [7] Documentación MySQL. *14.7.5.30 SHOW PROFILE Syntax*. Recuperado el 4 de diciembre de 2016, desde: <http://dev.mysql.com/doc/refman/5.7/en/show-profile.html>.
- [8] Documentación Nagios. *Nagios Core – Installing on Centos 7*. Recuperado el 7 de diciembre de 2016, desde: <https://assets.nagios.com/downloads/nagioscore/docs/Nagios-Core-Installing-On-Centos7.pdf>.
- [9] Página oficial de Munin. *Demo Munin-Monitoring: diskstats-iops*. Recuperado el 4 de diciembre de 2016, desde: http://demo.munin-monitoring.org/munin-monitoring.org/demo.munin-monitoring.org/diskstats_iops/sda.html.
- [10] Página oficial de Munin. *Demo Munin-Monitoring: load*. Recuperado el 4 de diciembre de 2016, desde: <http://demo.munin-monitoring.org/munin-monitoring.org/builddd.munin-monitoring.org/load.html>.
- [11] Página oficial de Munin. *Demo Munin-Monitoring: processes*. Recuperado el 4 de diciembre de 2016, desde: <http://demo.munin-monitoring.org/munin-monitoring.org/builddd.munin-monitoring.org/processes.html>.

- [12] Josef Weidendorfer. *KCachegrind*. Recuperado el 5 de diciembre de 2016, desde: <https://kcachegrind.github.io/html/Home.html>.