

Ingeniería de Servidores (2015-2016)
GRADO EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA



Práctica 3

José Carlos Martínez Velázquez

1 de diciembre de 2015

Índice

1. a) ¿Qué archivo le permite ver qué programas se han instalado con el gestor de paquetes? b) ¿Qué significan las terminaciones .1.gz o .2.gz de los archivos en ese directorio? 4
2. ¿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). 4
3. 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. 5
4. Ejecute el monitor de "System Performance" y muestre el resultado. Incluya capturas de pantalla comentando la información que aparece. 6
5. 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. 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. 11
7. 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. 14
8. Escriba un breve resumen sobre alguno de los artículos donde se muestra el uso de strace o busque otro y coméntelo. 16
9. 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). 17
10. Cuestión opcional 1: Indique qué comandos ha utilizado para realizarlo así como capturas de pantalla del proceso de reconstrucción del RAID. 20

- 11. Cuestión opcional 2: 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

1. a) ¿Qué archivo le permite ver qué programas se han instalado con el gestor de paquetes? b) ¿Qué significan las terminaciones .1.gz o .2.gz de los archivos en ese directorio?

- a) El archivo que permite ver qué programas se han instalado se encuentra en /var/-log/apt y se llama history.log. El gestor de paquetes yum permite ver su histórico a través del comando:

```
:~ $yum history
```

- b) Cada vez que hacemos una instalación, se escribe en un log, por lo que es lógico que los directorios de log crezcan cada vez más, de hecho si hiciéramos muchas instalaciones, los directorios de log serían inmanejables. Para controlar ésto, se implementa en /etc/cron.daily (tareas que se llevan a cabo diariamente) el script logrotate (o rotación de logs), que de manera programable comprime los logs y los almacena. Cada vez que actúa, crea un paquete nuevo, con extensión "1.gz", y suma uno a la terminación de todos los demás, de manera que cuanto mayor es el número de la extensión, más antiguo es el log. Por ejemplo si teníamos history.1.gz e history.2.gz, cuando actúe logrotate tendremos history.1.gz, history.2.gz e history.3.gz de manera que éstos dos últimos eran los que ya teníamos.

(Fuentes: [13] [12] [23])

2. ¿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).

El archivo que se debe modificar es /etc/crontab, que contiene todos los comandos que se ejecutan automáticamente mediante un periodo de tiempo. Una entrada de crontab tiene la siguiente sintaxis:

```
# m h dom mon dow user command
```

Donde:

- m: minuto en el que se ejecuta el comando
- h: hora en la que se ejecuta el comando
- dom: número de día en que se ejecuta el comando (1-31)

- mon: mes en que se ejecuta el comando (1-12)
- dow: día de la semana en que se ejecuta el comando (mon-sun)
- user: usuario que ejecutará el comando
- command: comando a ejecutar

Si cualquiera de los campos puede contener "cualquier" basta con poner el carácter '*' en la parte correspondiente.

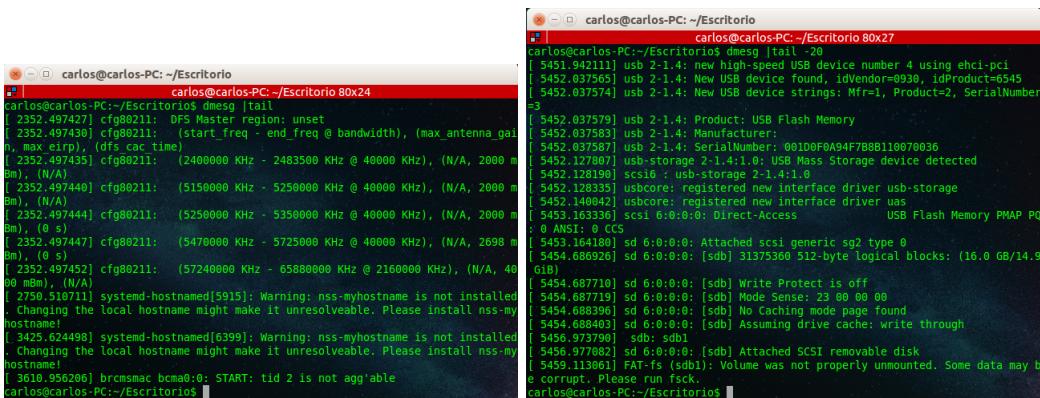
La línea necesaria para realizar una copia de seguridad de /codigo a /seguridad/\$fecha una vez al día sería:

```
00 00 * * * carlos cp /home/carlos/codigo /home/carlos/seguridad/"`date`"
```

(Fuentes: [3] [20])

3. 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.

La herramienta dmesg (diagnostic message) sirve para listar mensajes del buffer del nucleo, es decir, mensajes que se generan según van ocurriendo eventos. En este caso, vamos a ejecutar la utilidad antes y después de introducir un pen drive. Obtenemos lo siguiente:



```
carlos@carlos-PC: ~/Escritorio
carlos@carlos-PC: ~/Escritorio 80x27
[ 2352.497427] cfg80211: DFS Master region: unset
[ 2352.497438] cfg80211: (start_freq - end_freq @ bandwidth), (max_antenna_gain, max_eirp), (dfs_cac_time)
[ 2352.497435] cfg80211: (2400000 KHz - 2483500 KHz @ 40000 KHz), (N/A, 2000 mBm), (N/A)
[ 2352.497440] cfg80211: (5150000 KHz - 5250000 KHz @ 40000 KHz), (N/A, 2000 mBm), (N/A)
[ 2352.497444] cfg80211: (5250000 KHz - 5350000 KHz @ 40000 KHz), (N/A, 2000 mBm), (0 s)
[ 2352.497447] cfg80211: (5470000 KHz - 5725000 KHz @ 40000 KHz), (N/A, 2698 mBm), (0 s)
[ 2352.497452] cfg80211: (57240000 KHz - 65880000 KHz @ 2160000 KHz), (N/A, 4000 mBm), (N/A)
[ 2750.510711] systemd-hostnamed[5915]: Warning: nss-myhostname is not installed
[ 2750.510711] systemd-hostnamed[5915]: Changing the local hostname might make it unresolvable. Please install nss-myhostname
[ 3425.624498] systemd-hostnamed[6399]: Warning: nss-myhostname is not installed
[ 3425.624498] systemd-hostnamed[6399]: Changing the local hostname might make it unresolvable. Please install nss-myhostname
[ 3610.956296] brcmssma hhma0: START: tid 2 is not agg'able
carlos@carlos-PC:~/Escritorio$ 
```

(a) Antes de conectar el dispositivo.

(b) Despues de conectar el dispositivo.

Figura 3.1: Utilización de dmesg.

Lo primero que observamos, es que los mensajes en cada una de las dos ejecuciones son diferentes. Antes de introducir el pen drive, los últimos mensajes que se generaron

tuvieron que ver previsiblemente con la red inalámbrica, ya que aparecen frecuencias medidas en KHz, palabras como bandwidth, antenna, hostname, etc. En la segunda ejecución, ya sí que vemos que se menciona la interfaz USB, por lo que, a priori, tiene que ver con lo que hemos hecho.

Como podemos ver, dmesg nos ofrece información bastante útil, tal como el tipo de dispositivo, el ID del vendedor y del producto así como el número de serie, quizás irrelevantes para el usuario, pero de cara a una avería, son datos necesarios. Otra información interesante es con qué interfaz ha montado el dispositivo, en éste caso SCSI y dónde (/dev/sdb). La información extra acerca del estado del hardware conectado viene dada por la protección contra escritura (desactivada), la página de mode sense [8], la página de caching mode, y el tipo de escritura en el dispositivo (escritura directa - write through).

(Fuentes: [8] [21])

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

El monitor de rendimiento del sistema nos ofrece información de diversos tipos. Puede ser clasificada en los siguientes apartados:

- Disco físico.
- Información del procesador.
- Interfaz de red.
- Memoria.

El resultado de la ejecución en mi máquina virtual es el siguiente:

\\WIN-30EDAC4EN1A			
Disco físico	Total	0 C:	1 E:
% de tiempo inactivo	100,105	100,098	100,108
Longitud promedio de la cola de disco	0,000	0,000	0,000

(a) Información del disco físico.

Información del procesador	Total	0_C_Total	0_E_Total
% de tiempo de interrupción	0,000	0,000	0,000
% de tiempo de procesador	0,062	0,062	0,062
Estado de detención	0,000	0,000	0,000

(b) Información del procesador.

Interfaz de red	Adaptador de escritorio Intel[R] PRO_1000 MT
Total de bytes/s	0,000
<hr/>	
Memoria	
% de bytes confirmados en uso	18,508
Errores de caché/s	0,000
Mbytes disponibles	615.000

(c) Información de red y memoria.

Figura 4.1: Monitor de rendimiento del sistema de Windows Server 2012 r2.

Del apartado de disco físico, podemos deducir que en el instante de la captura, el sistema estaba prácticamente ocioso. En el apartado de información del procesador, podemos deducir que en el instante de la captura no hubo interrupciones ni detenciones, pero el monitor del sistema en su time-slice ocupaba un 0,062 % del procesador. En el apartado de red, podemos ver que el dispositivo no estaba conectado o bien tenía una velocidad de 0b/s. En el apartado de memoria deducimos que se estaban usando un 18,5 % de los bites disponibles, que no hubo errores de caché y que aún quedaban 615 Mb de RAM libres. Además podemos ver el gráfico que genera el monitor de rendimiento:

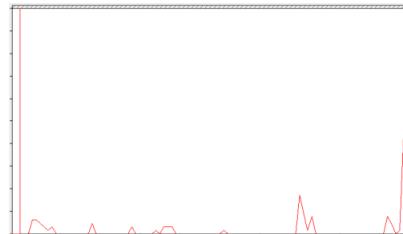


Figura 4.2: Gráfico del monitor de rendimiento.

(Fuentes: [16] [17])

- 5. 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.**

Para crear un conjunto recopilador de datos hay que entrar en el monitor de rendimiento, en la parte izquierda clicar en conjuntos de recopiladores de datos, posteriormente clic derecho en definidos por el usuario y pulsamos Nuevo → Conjunto de recopiladores de datos. Se abre un asistente. Le damos un nombre y elegimos el modo avanzado.

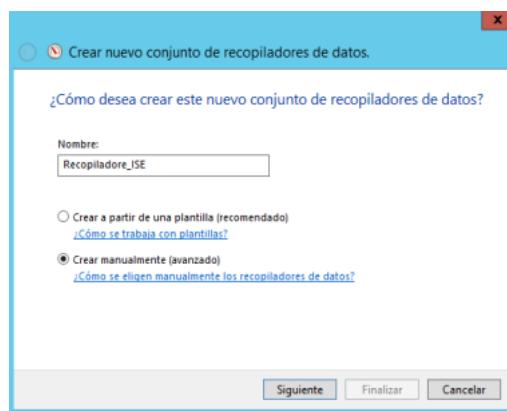


Figura 5.1: Cración de un conjunto de recopiladores de datos.

En el siguiente paso seleccionamos el contador de rendimiento como los datos de seguimiento.

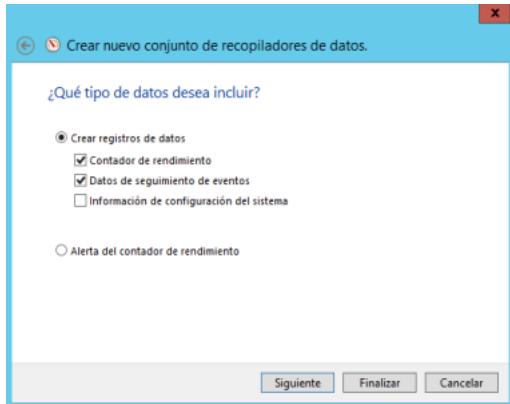
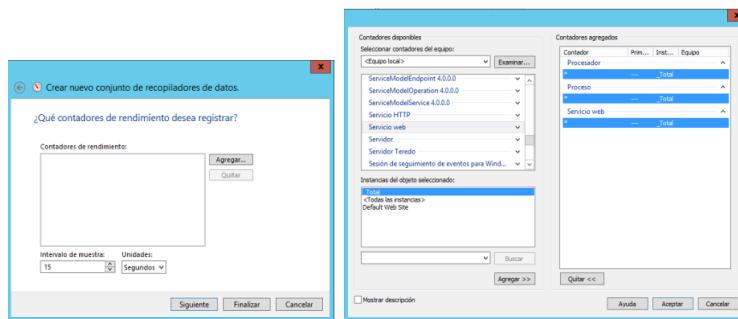
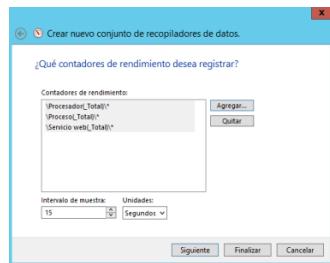


Figura 5.2: Selección de los contadores de rendimiento que incluye.

En el siguiente paso seleccionamos el tipo de contadores de rendimiento que queremos que incluya. Para ello clicamos en Agregar y seleccionamos los tipos de contadores de rendimiento que queremos incluir, en éste caso, todos los referentes al procesador, al proceso y al servicio web. Una vez hecho, aparecerán en la pantalla anterior. Seleccionamos el intervalo de 15 segundos y podemos clicar en Finalizar.



(a) Pantalla de tipos de contadores de rendimiento. (b) Seleccionando los tipos de contadores de rendimiento.



(c) Los tipos de contadores de rendimiento seleccionados han sido incluidos.

Figura 5.3: Selección del tipo de contadores de rendimiento.

Para cambiar la ruta del informe, avanzamos dos páginas en el asistente y escribimos la ruta del escritorio:clic derecho en el recopilador (parte izquierda) y seleccionamos administrador de datos. En el textfield Nombre del archivo del informe, tecleamos la ruta "C:\Users\Administrador\Desktop\logs

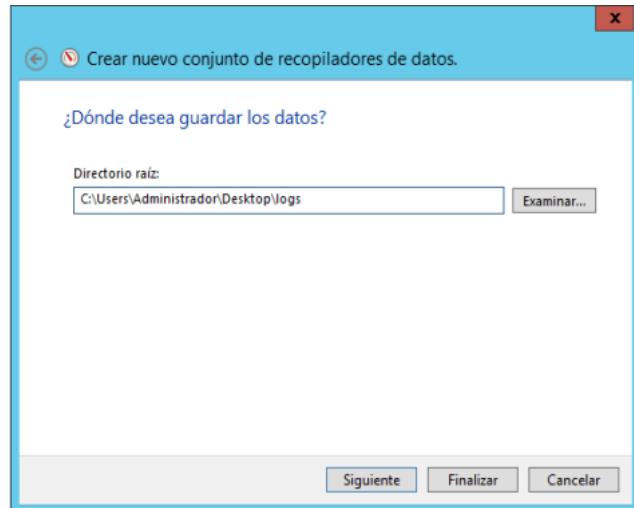


Figura 5.4: Cambiando la ruta de logs a Escritorio\logs.

Una vez hemos creado el recopilador de datos, lo ejecutamos (botón triángulo verde) y esperamos un rato. Mientras tanto hemos solicitado la página home de IIS desde el navegador anfitrión. Éste es el resultado:

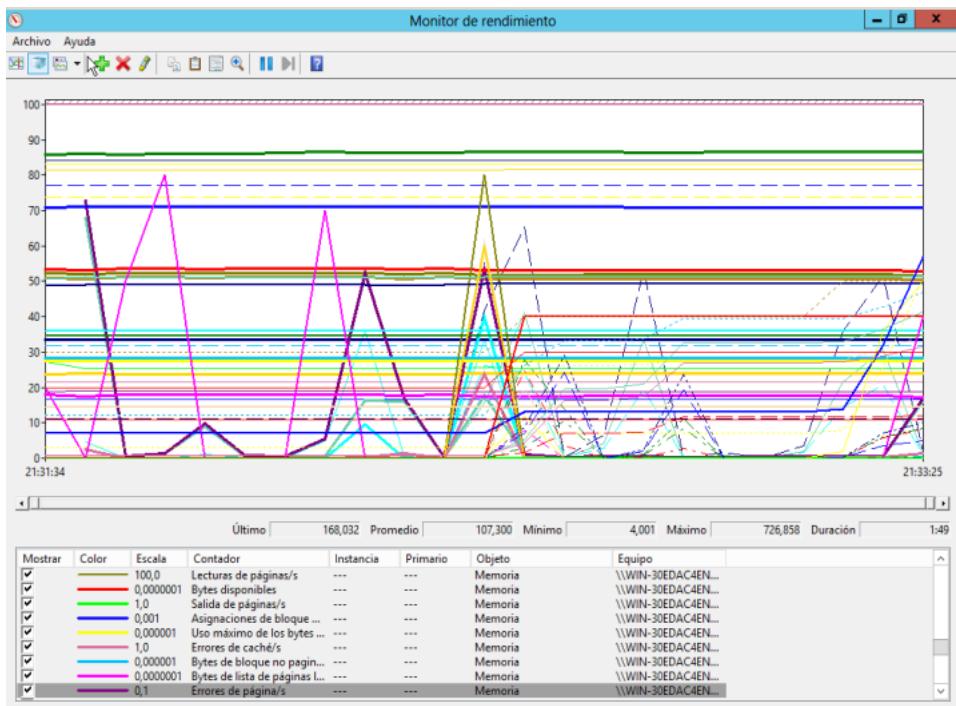


Figura 5.5: Ejecución del recopilador de datos.

(Fuentes: [15])

6. 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.

En un principio, quería probar xsensors (la versión con GUI de lm-sensors) en la máquina real, para obtener datos realistas, pero su interfaz sólo ofrece los datos de temperatura del core y del disco duro, lo que me pareció una información muy pobre:



Figura 6.1: Interfaz de xsensors.

Encontré que se podía ver los sensores que contenía la máquina, lo que respondió a mi pregunta. No era que la interfaz de xsensors fuera poco potente, sino que mi máquina carecía de más sensores. Para saberlo ejecuté el comando:

```
:~ $sudo sensors-detect
```

Y obtuve el siguiente resultado:

```
carlos@carlos-PC: ~ carlos@carlos-PC: ~ 80x22
Some south bridges, CPUs or memory controllers contain embedded sensors.
Do you want to scan for them? This is totally safe. (YES/no): YES
Module cpuid loaded successfully.
Silicon Integrated Systems SIS5595... No
VIA VT82C686 Integrated Sensors... No
VIA VT8231 Integrated Sensors... No
AMD K8 thermal sensors... No
AMD Family 10h thermal sensors... No
AMD Family 11h thermal sensors... No
AMD Family 12h and 14h thermal sensors... No
AMD Family 15h thermal sensors... No
AMD Family 15h power sensors... No
AMD Family 16h power sensors... No
Intel digital thermal sensor... Success!
  (driver 'coretemp')
Intel AMB FB-DIMM thermal sensor... No
VIA C7 thermal sensor... No
VIA Nano thermal sensor... No

Some Super I/O chips contain embedded sensors. We have to write to
standard I/O ports to probe them. This is usually safe.
Do you want to scan for Super I/O sensors? (YES/no):
```

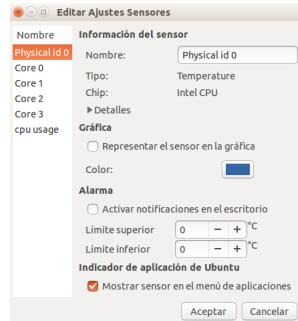
Figura 6.2: Sensores disponibles en mi máquina.

Una alternativa que me ha llamado la atención, es Psensor. Una herramienta que

permite monitorizar gráficamente la temperatura con los sensores disponibles, además de seleccionar qué aparece y qué no en la gráfica así como personalizar cada sensor con colores, máxima temperatura, etc.



(a) Pantalla principal de Psensor.



(b) Personalizando el sensor del core 0.

Figura 6.3: Usando Psensor.

(Fuentes: [22] [14])

- 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.

Vamos a analizar algunas de las gráficas que aquí aparecen. La primera será el uso de disco cada día.

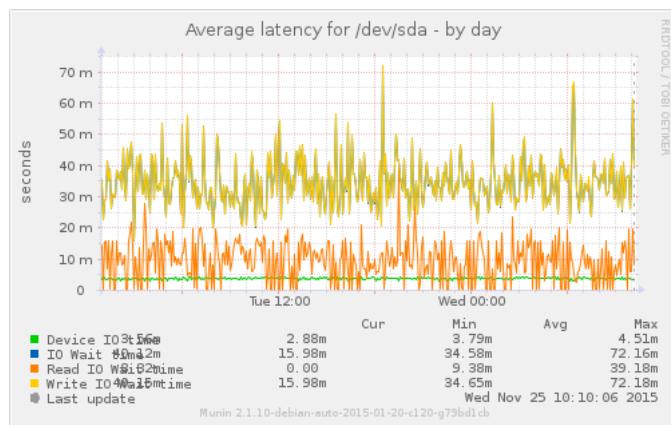


Figura 7.1: Uso de disco diariamente.

Como podemos observar, la entrada y salida (verde) se mantiene constante por lo que no hay anomalías evidentes. Tanto las lecturas como las escrituras tienen altibajos muy evidentes, pero se podría deducir que entre las 18:00 y las 20:00 horas se encuentra el mayor tráfico, esto es por los picos que encontramos en naranja y amarillo respectivamente. Más adelante podemos ver más picos similares, lo que podría tratarse de copias de seguridad o escrituras en log periódicas programadas.

La siguiente gráfica que analizaremos será el número de usuarios on-line a la semana.

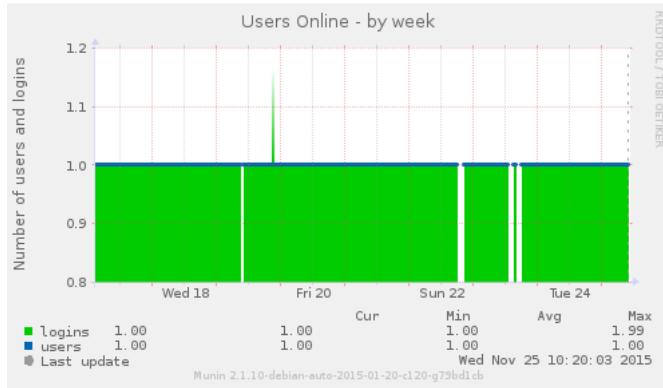


Figura 7.2: Usuarios on-line a la semana.

Esta gráfica es bastante homogénea, por lo que se puede deducir la carga habitual (al menos de esa semana) muy fácilmente, de cara a dotar a nuestro servidor de la capacidad adecuada de procesamiento, tiempo de respuesta y productividad. El jueves 19 así como el domingo 22 y el lunes 23 podemos observar interrupciones muy evidentes, lo que podría de tratarse de interrupciones del servicio por mantenimiento o caídas del sistema. Dado que se encuentran muy juntas, sobre todo las de domingo y lunes, me decantaría por la opción de que se tratase de tareas de mantenimiento, pero habría que consultar los logs del sistema. También podemos ver como el día 19, a últimas horas, encontramos un pico de logins de clientes no habitual, lo que podría tratarse de una promoción (en caso de una tienda on-line), programa especial (en caso de un servidor streaming), un suceso de gran relevancia (en caso de un periódico digital o canal de noticias) o, en definitiva, cualquier evento no habitual.

Por último, analizaremos el uso de la red al cabo del mes.

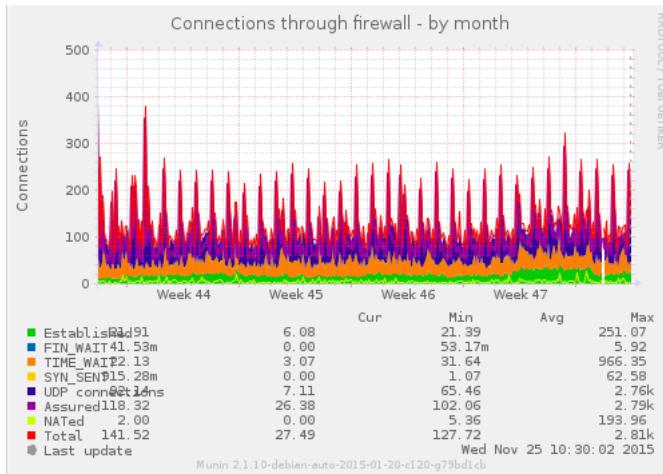


Figura 7.3: Uso de red al mes.

Podemos ver que por cada semana, se aprecian siete picos muy evidentes y diferencias entre ellos, por lo que podemos deducir que en una franja horaria muy determinada es cuando tiene el máximo tráfico y no es una carga homogénea. Por curiosidad, decidí mirar la gráfica de uso diario.

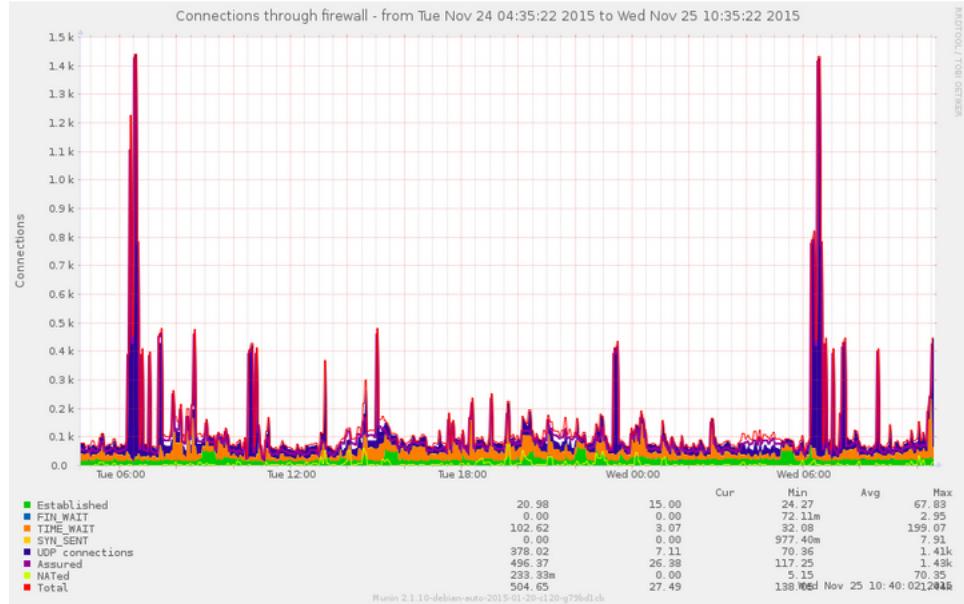


Figura 7.4: Uso diario de la red.

Podemos ver que el pico de uso diario es por la mañana, más concretamente entre las 6:00 y las 9:00 por lo que, probablemente se trate de un periódico digital.

(Fuentes: [9])

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

El artículo [1] habla acerca de cómo llegó a las manos del narrador un servidor SVN que no funcionaba y consiguió que se comportara correctamente con strace, una herramienta que permite monitorizar las llamadas a sistema, de ahí su nombre: system call tracer.

La sintaxis básica de strace es:

```
strace [opciones] program [argumentos_del_programa]
```

Para hacer una prueba, se ejecuta strace /usr/bin/uptime, lo que devuelve todas las llamadas que hace el sistema para ejecutar uptime. Los mensajes de strace pueden ser de tres tipos:

- Apertura de archivos.
- Códigos que devuelven las llamadas que fallan.
- La salida de texto por consola.

Dos opciones muy comunes de strace son:

- -o, que sirve para guardar el resultado de la ejecución de strace en un archivo, en definitiva, redirigir la salida. se usa:

```
strace -o [archivo_al_que_redirigir] [programa_que_se_ejecuta]
```

- -p, que permite conectarte a un programa que se está ejecutando a través de su PID. Es útil para monitorizar demonios.

A continuación de ésto, el narrador explica los detalles concretos de cómo solucionó el problema, cosa irrelevante para el funcionamiento de strace.

El otro artículo, [6], explica qué es auditar un programa. En definitiva, si tuvieramos que auditar un programa, lo que nos están pidiendo es que emitamos un informe hablando acerca de lo que hace el programa en cuestión, pero ¿cómo hacerlo? para ello tenemos la herramienta strace. En el artículo, el autor decide auditar a modo de ejemplo un "hola mundo" utilizando strace. Este tipo de programa hace dos llamadas a sistema: fstat y write. La llamada fstat ofrece información acerca del estado del archivo (el fichero que contiene el programa), mientras que write saca el mensaje por pantalla. Posteriormente el autor deja entrever que strace no es la panacea, ya que no permite, por ejemplo, realizar trazas de tiempo.

(Fuentes: [1] [6])

9. 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).

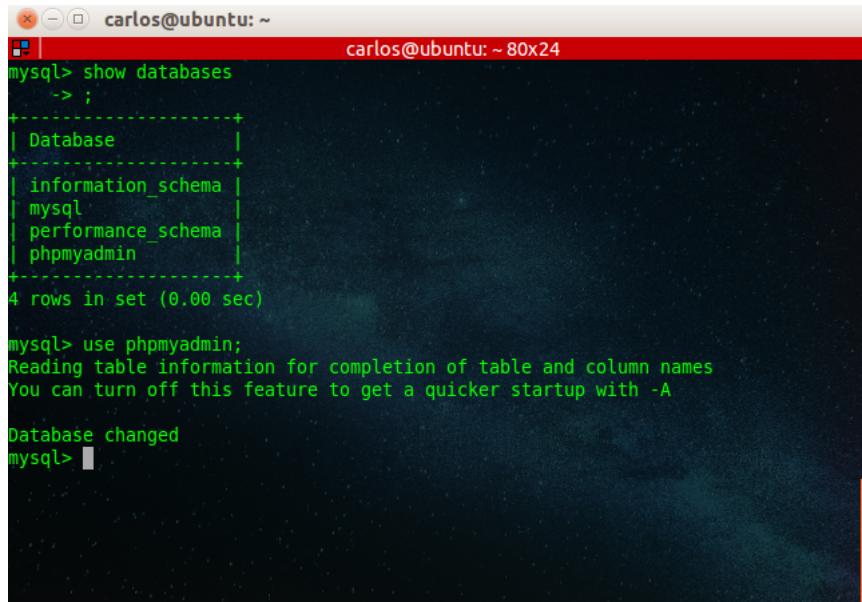
Accedemos a mysql con el comando:

```
:~ $mysql -u root -p
```

Ingresando la contraseña especificada durante la elaboración de la práctica 2. Utilizaremos la base de datos de PhpMyAdmin, para ello usaremos los comandos:

```
>show databases;
>use phpmyadmin;
```

Tal como se muestra en la siguiente figura.



```
carlos@ubuntu: ~
carlos@ubuntu: ~ 80x24
mysql> show databases
    -> ;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| phpmyadmin     |
+-----+
4 rows in set (0.00 sec)

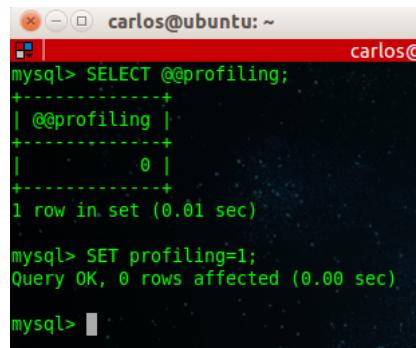
mysql> use phpmyadmin;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>
```

Figura 9.1: Accediendo a la base de datos PhpMyAdmin.

Antes de hacer nada, vamos a indicar que queremos hacer profiling a la sesión, así que ejecutaremos la siguiente secuencia de comandos.

```
>SELECT @@profiling;
>SET profiling=1;
```



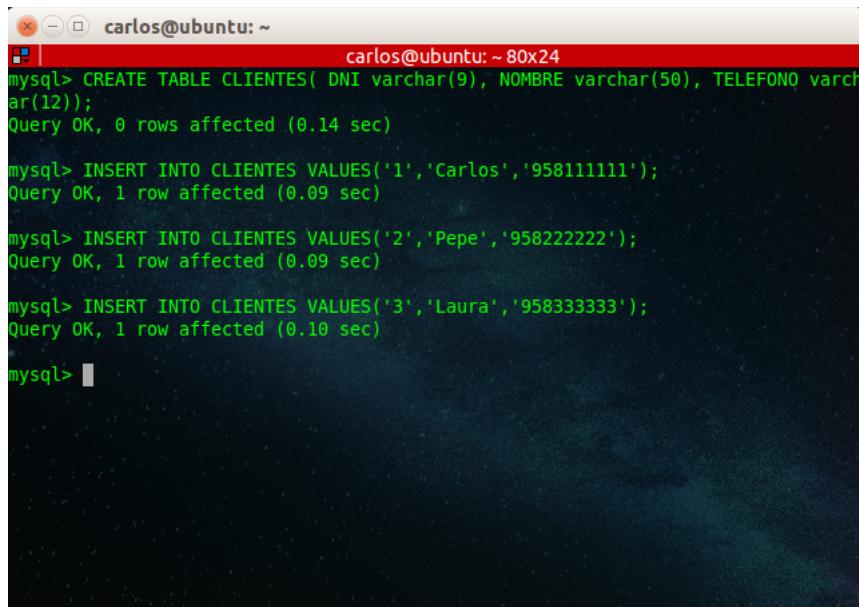
```
carlos@ubuntu: ~
carlos@ubuntu: ~ 80x24
mysql> SELECT @@profiling;
+-----+
| @@profiling |
+-----+
|          0 |
+-----+
1 row in set (0.01 sec)

mysql> SET profiling=1;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Figura 9.2: Configuración de profiling.

Una vez hecho ésto, vamos a crear una tabla, que llamaremos CLIENTES, a la que introduciremos algunos registros:



A screenshot of a terminal window titled "carlos@ubuntu: ~". The window shows MySQL commands being run:

```
carlos@ubuntu: ~ 80x24
mysql> CREATE TABLE CLIENTES( DNI varchar(9), NOMBRE varchar(50), TELEFONO varchar(12));
Query OK, 0 rows affected (0.14 sec)

mysql> INSERT INTO CLIENTES VALUES('1','Carlos','958111111');
Query OK, 1 row affected (0.09 sec)

mysql> INSERT INTO CLIENTES VALUES('2','Pepe','958222222');
Query OK, 1 row affected (0.09 sec)

mysql> INSERT INTO CLIENTES VALUES('3','Laura','958333333');
Query OK, 1 row affected (0.10 sec)

mysql> |
```

Figura 9.3: Creando tablas y registros.

Una vez hecho, podemos ejecutar los siguientes comandos:

```
>SHOW PROFILES;
>SHOW PROFILE;
>SHOW PROFILE FOR QUERY n;
```

Donde, el primero sirve para ver cuánto duró la ejecución de cada comando. El segundo, sirve para ver cuánto tiempo se invirtió en cada tarea (comprobar permisos, ejecutar la orden, reunir los datos, impresión por pantalla, etc). El tercero nos dará la misma información que el segundo pero para una consulta 'n' concreta. Probémoslo.

```
carlos@ubuntu:~  
[1] 170x41  
mysql> SELECT * FROM CLIENTES WHERE DNI='1';  
+-----+-----+-----+  
| DNI | NOMBRE | TELEFONO |  
+-----+-----+-----+  
| 1 | Carlos | 958111111 |  
+-----+-----+-----+  
1 row in set (0.00 sec)  
  
mysql> SHOW PROFILE;  
+-----+-----+-----+  
| Query_ID | Duration | Query |  
+-----+-----+-----+  
| 1 | 0.12591700 | CREATE TABLE CLIENTES( DNI varchar(9), NOMBRE varchar(50), TELEFONO varchar(12)) |  
| 2 | 0.10863850 | INSERT INTO CLIENTES VALUES('1','Carlos','958111111') |  
| 3 | 0.09735275 | INSERT INTO CLIENTES VALUES('2','Pepe','958222222') |  
| 4 | 0.08994125 | INSERT INTO CLIENTES VALUES('3','Laura','958333333') |  
| 5 | 0.00077950 | SELECT * FROM CLIENTES WHERE DNI='1' |  
+-----+-----+-----+  
5 rows in set (0.00 sec)
```

(a) Consulta y SHOW PROFILES.

```
carlos@ubuntu:~  
[1] 170x41  
mysql> SHOW PROFILE;  
+-----+-----+  
| Status | Duration |  
+-----+-----+  
| starting | 0.000035 |  
| waiting for query cache lock | 0.000006 |  
| checking query cache for query | 0.000069 |  
| checking permissions | 0.000013 |  
| opening tables | 0.000032 |  
| system lock | 0.000014 |  
| waiting for query cache lock | 0.000034 |  
| init | 0.000046 |  
| optimizing | 0.000018 |  
| statistics | 0.000023 |  
| preparing | 0.000023 |  
| executing | 0.000005 |  
| sending data | 0.000094 |  
| end | 0.000011 |  
| query end | 0.000007 |  
| closing tables | 0.000011 |  
| freeing items | 0.000012 |  
| waiting for query cache lock | 0.000004 |  
| freeing items | 0.000029 |  
| waiting for query cache lock | 0.000010 |  
| freeing items | 0.000003 |  
| storing result in query cache | 0.000007 |  
| logging slow query | 0.000004 |  
| cleaning up | 0.000006 |  
+-----+-----+  
24 rows in set (0.00 sec)
```

(b) SHOW PROFILE.

```
mysql> SHOW PROFILE FOR QUERY 1;  
+-----+-----+  
| Status | Duration |  
+-----+-----+  
| starting | 0.000099 |  
| checking permissions | 0.000019 |  
| opening tables | 0.000107 |  
| system lock | 0.000045 |  
| creating table | 0.124793 |  
| after create | 0.000019 |  
| query end | 0.000007 |  
| closing tables | 0.000017 |  
| freeing items | 0.000079 |  
| logging slow query | 0.000013 |  
| cleaning up | 0.000009 |  
+-----+-----+  
11 rows in set (0.01 sec)
```

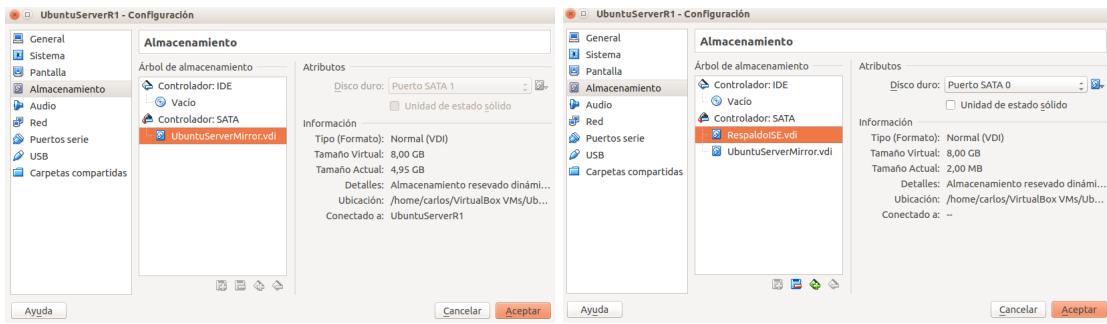
(c) SHOW PROFILE FOR QUERY 1.

Figura 9.4: Consultando profiles de MySQL.

(Fuentes: [2] [10])

10. Cuestión opcional 1: Indique qué comandos ha utilizado para realizarlo así como capturas de pantalla del proceso de reconstrucción del RAID.

Para recuperar el RAID1 de la práctica 1, lo primero que haremos es conectar un disco nuevo en la máquina. Para ello, debemos crear una unidad virtual idéntica a la que ya teníamos.



(a) Propiedades del disco anterior.

(b) Propiedades del nuevo disco.

Figura 10.1: Creando nuevo disco para configurar RAID1.

Ya tenemos todo lo necesario para empezar. Arrancamos la máquina. Ahora cuidado, hay que seleccionar el disco que arranca el sistema operativo y para ello pulsaremos F12 durante el arranque. Seleccionamos el disco 2, que era el que ya teníamos antes y comenzará a cargar el S.O. con normalidad.

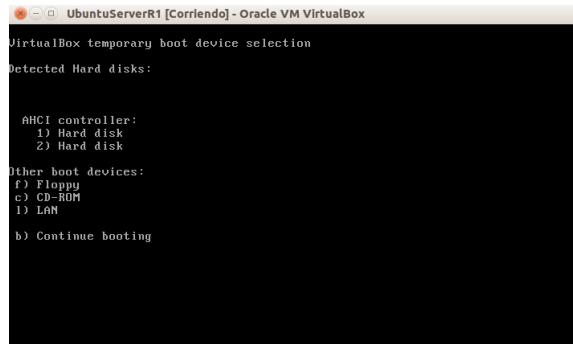


Figura 10.2: Seleccionando el disco arrancable.

Si ejecutamos fdisk -l, obtendremos información de los discos. En la siguiente figura podemos observar cómo tenemos un disco que no es arrancable (/dev/sda) y otro que sí lo es (/dev/sdb). El primero, que no es arrancable, es nuestro nuevo disco.

```
UbuntuServerR1 [Corriendo] - Oracle VM VirtualBox
GNU nano 2.2.6           Archivo: fdiskl.txt

Disco /dev/sda: 8589 MB, 8589934592 bytes
255 cabezas, 63 sectores/pista, 1044 cilindros, 16777216 sectores en total
Unidades = sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico / físico): 512 bytes / 512 bytes
Tamaño E/S (mínimo/óptimo): 512 bytes / 512 bytes
Identificador del disco: 0x00000000

Disco /dev/sdb: 8589 MB, 8589934592 bytes
255 cabezas, 63 sectores/pista, 1044 cilindros, 16777216 sectores en total
Unidades = sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico / físico): 512 bytes / 512 bytes
Tamaño E/S (mínimo/óptimo): 512 bytes / 512 bytes
Identificador del disco: 0x00005fd4

Dispositivo Inicio     Comienzo    Fin      Bloques Id Sistema
/dev/sdb1      *        2048    16775167    8386560  fd  Linux raid autodetect
```

Figura 10.3: Consultando discos en la máquina.

Ahora consultaremos el archivo /proc/mdstat para saber qué multidevice es nuestro RAID1, como podemos observar es md0.

```
UbuntuServerR1 [Corriendo] - Oracle VM VirtualBox
carlos@ubuntu:~$ cat /proc/mdstat
Personalities : [raid1] [linear] [multipath] [raid0] [raid5] [raid4] [raid6]
md0 : active raid1 sdb1[1]
      8382400 blocks super 1.2 (Z/1) L_0
unused devices: <none>
carlos@ubuntu:~$ _
```

Figura 10.4: Consultando /proc/mdstat.

Antes de hacer nada, deberíamos darle formato a nuestro nuevo disco duro. Es decir, formatearlo en el mismo sistema de archivos que el disco original. Para consultar el sistema de archivos, ejecutamos el comando

```
:~$ df -T
```

La raiz, tiene el sistema de archivos ext4, algo que era de suponer.

Mount Point	Type	Total Blocks	Used Blocks	Available Blocks	File System Type	Mounted On
/	ext4	5632116	1395228	3927744	ext4	/
/dev/mapper/HDs-raiz_crypt	ext4	5632116	1395228	3927744	ext4	/
/dev/mapper/HDs-home_crypt	ext4	461056	2410	431074	ext4	/home
/home/carlos/.Private	ecryptfs	461056	2410	431074	ecryptfs	/home/carlos/.Private

Figura 10.5: Consultando /proc/mdstat cuando ya ha terminado la copia.

Damos formato al disco ejecutando:

```
:~$ sudo mkfs.ext4 /dev/sda
```

Una vez hecho, clonamos el contenido de /dev/sdb a /dev/sda con el comando:

```
:~$ sudo dd if=/dev/sdb | pv | sudo dd of=/dev/sda
```

```
UbuntuServerR1 [Corriendo] - Oracle VM VirtualBox
carlos@ubuntu:~$ sudo dd if=/dev/sdb | pv | sudo dd of=/dev/sda
64MB 0:01:45 [6,11MB/s] =>
```

Figura 10.6: Clonando /dev/sdb en /dev/sda.

Cuando termina la copia, podemos continuar con el proceso.

```

carlos@ubuntu:~$ sudo dd if=/dev/sdb of=/dev/sda
16772216+0 registros leídos
16772216+0 registros escritos
5699934592 bytes (0,6 GB) copiados, 569,171 s, 15,1 MB/s
0:09:29 (14,4MB/s)
16772216+0 registros leídos
16772216+0 registros escritos
5699934592 bytes (0,6 GB) copiados, 569,767 s, 15,1 MB/s
carlos@ubuntu:~_

```

Figura 10.7: Ha terminado la clonación de /dev/sdb en /dev/sda.

(Nota: En algunos casos es necesario instalar grub después de clonar porque si no, se queda bloqueado tras introducir la primera contraseña de desencriptación durante el arranque. Para ello usamos el comando grub-install, tal y como se muestra en la siguiente figura)

```

carlos@ubuntu:~$ sudo grub-install /dev/sdb
Instalando para plataforma i386-pc.
Instalación terminada. Ningún error encontrado.
carlos@ubuntu:~$

```

Figura 10.8: Instalando grub en el nuevo disco.

Ya tenemos todos los datos y el nuevo disco es una copia exacta de /dev/sdb. Vamos a añadir el disco /dev/sda al RAID md0. Para ello ejecutamos el comando:

```
:~ $ sudo mdadm /dev/md0 --add /dev/sda
```

(Nota: Si mdadm no está instalado debes instalarlo con sudo apt-get install mdadm).

```

carlos@ubuntu:~$ sudo mdadm /dev/md0 --add /dev/sda
mdadm: added /dev/sda
carlos@ubuntu:~$

```

Figura 10.9: Añadiendo el disco al RAID.

En este momento parece que no hace nada, pero si ejecutamos cat /proc/mdstat podemos observar cómo en diferentes instantes se va completando la recuperación. En éste momento, está copiando los datos desde /dev/sdb a /dev/sda. Hay que ir consultando periódicamente para saber cuándo ha terminado:

```

carlos@ubuntu:~$ cat /proc/mdstat
Personalities : (raid1) [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sda(2) sdb(1)
      8382400 blocks super 1.2 [2:1] [_U]
      [======>.....] recovery = 30.8x (2582656/8382400) finish=3.1min
      speed=30615K/sec

unused devices: <none>
carlos@ubuntu:~$ cat /proc/mdstat
Personalities : (raid1) [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sda(2) sdb(1)
      8382400 blocks super 1.2 [2:1] [_U]
      [======>.....] recovery = 31.5x (2647936/8382400) finish=3.0min
      speed=30910K/sec

unused devices: <none>
carlos@ubuntu:~_

```

Figura 10.10: Consultando /proc/mdstat en dos instantes de tiempo distintos durante la recuperación.

En un momento determinado, consultaremos /proc/mdstat y ya no saldrán porcentajes ni barras de progreso. En este instante, podemos saber que ya ha acabado la copia y que, además, el RAID1 (/dev/md0) se compone de dos discos, pues vemos que pone [2/2], lo que además nos indica que ambos están operativos. Fijémonos en la diferencia entre la figura 10.11 y la figura 10.4

```
UbuntuServerR1 [Corriendo] - Oracle VM VirtualBox
carlos@ubuntu:~$ cat /proc/mdstat
Personalities : (raid1) (linear) (multipath) (raid0) (raid1) (raid5) (raid4) (raid10)
md0 : active raid1 sda1[1] sdb1[0]
      8382400 blocks super 1.2 [2/2] [UU]
unused devices: <none>
carlos@ubuntu:~$
```

Figura 10.11: Consultando /proc/mdstat cuando ya ha terminado la copia.

Para comprobar que el RAID1 está funcionando correctamente, bien podríamos desconectar el disco /dev/sdb, pero ésto nos supondría volver a pasar por éste proceso y volverlo a comprobar, lo que no parece efectivo. Al principio del proceso, teníamos que pulsar F12 para escoger el disco que debía arrancar, ya que en caso contrario, el sistema no arrancaba, al intentar arrancar con /dev/sda. Si reiniciamos el sistema y ésto no ocurre, podríamos determinar que el RAID1 ahora está correctamente configurado. Antes de arrancar vamos a consultar de nuevo fdisk -l, donde podemos ver si /dev/sda es arrancable (Nótese la diferencia entre la figura 10.3 y la figura 10.13):

```
UbuntuServerR1 [Corriendo] - Oracle VM VirtualBox
GNU nano 2.2.6 Archivo: fdisk1.txt

Disco /dev/sda: 8589 MB, 858934592 bytes
255 cabezas, 63 sectores/pista, 1044 cilindros, 16777216 sectores en total
Unidades = sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico / físico): 512 bytes / 512 bytes
Tamaño E/S (mínimo/óptimo): 512 bytes / 512 bytes
Identificador del disco: 0x0005fada

Dispositivo Inicio Comienzo Fin Bloques Id Sistema
/dev/sda1 * 2048 16775167 8306560 fd Linux raid autodetect

Disco /dev/sdb: 8589 MB, 858934592 bytes
255 cabezas, 63 sectores/pista, 1044 cilindros, 16777216 sectores en total
Unidades = sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico / físico): 512 bytes / 512 bytes
Tamaño E/S (mínimo/óptimo): 512 bytes / 512 bytes
Identificador del disco: 0x0005fada

Dispositivo Inicio Comienzo Fin Bloques Id Sistema
/dev/sdb1 * 2048 16775167 8306560 fd Linux raid autodetect
```

Figura 10.12: Comprobando que /dev/sda ahora es arrancable.

Si reiniciamos:

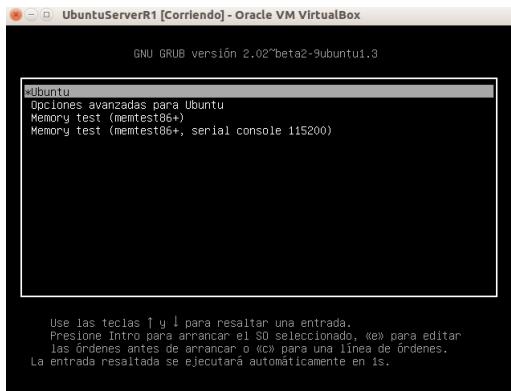


Figura 10.13: El sistema arranca sin seleccionar disco.

¡Éxito!

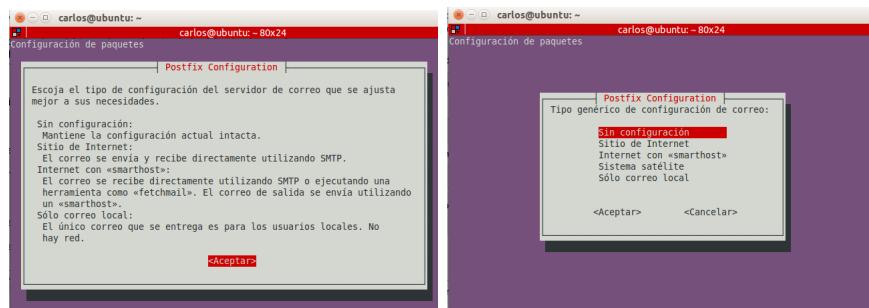
(Fuentes: [11] [19] [18] [7] [4])

11. Cuestión opcional 2: instale Nagios en su sistema (el que prefiera) documentando el proceso y muestre el resultado de la monitorización de su sistema comentando qué aparece.

Vamos a instalar Nagios. Para ello, vamos a ejecutar el siguiente comando:

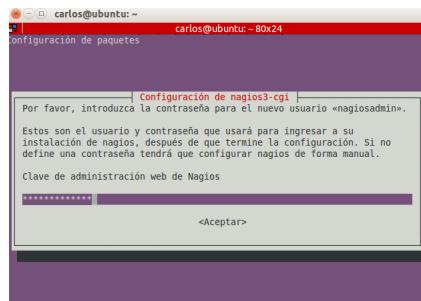
```
:~$sudo apt-get install nagios3 nagios-nrpe-plugin
```

Nos saldrán la siguiente pantalla, de elección del tipo de configuración del servidor. Dado que no tenemos definido qué servicio prestaremos, elegiremos sin configuración. Se nos solicitará definir una contraseña y posteriormente confirmarla. Luego, continúa la instalación.



(a) Tipo de configuración del servidor

(b) Sin configuración



(c) Definiendo contraseña

Figura 11.1: Instalando Nagios.

Una vez ha terminado la instalación de nagios, vamos a ver cómo administrar los usuarios. Para crear un usuario, hay que agregarlo en el archivo /etc/nagios3/htpasswd.users. Esto se automatiza con el comando:

```
:~ $sudo htpasswd /etc/nagios3/htpasswd.users [ nombre_usuario ]
```

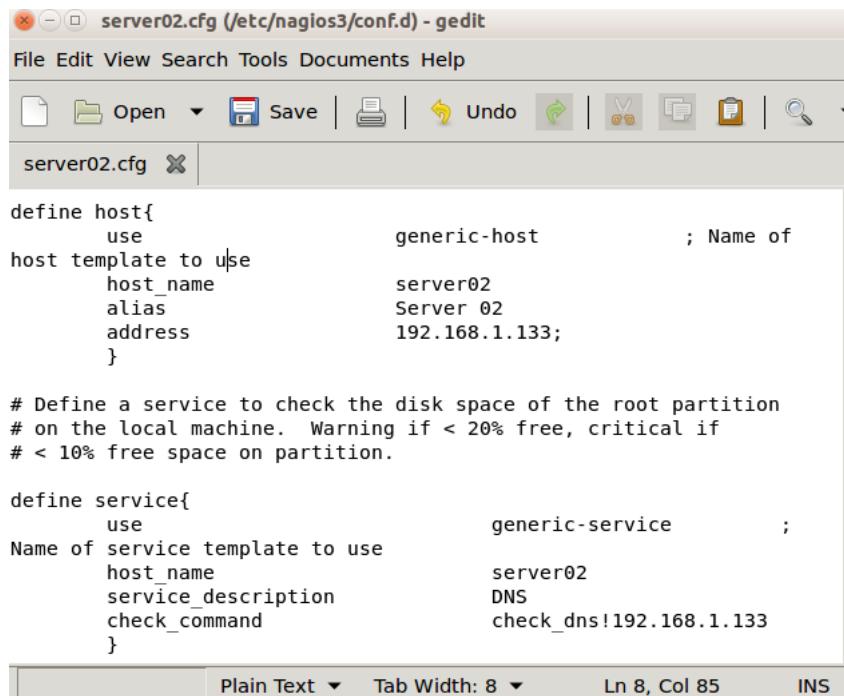
Para monitorizar en remoto, hay que instalar también el paquete nagios-nrpe-server en otra máquina (ésta sera server02, mi máquina real). Lo hacemos con:

```
:~ $sudo apt-get install nagios-nrpe-server
```

Crearemos un archivo de configuración de host. Para ello, ejecutamos:

```
:~ $sudo cp /etc/nagios3/conf.d/localhost_nagios2.cfg /etc/nagios3/conf.d/server02.cfg
```

Hay que editar el archivo /etc/nagios3/conf.d/server02.cfg para que quede del siguiente modo, siendo la IP que se ve la de server02:



The screenshot shows the gedit text editor with the file 'server02.cfg' open. The code defines a host named 'server02' with specific parameters like host_name, alias, and address. It also defines a service for this host using a generic-service template.

```
define host{
    use generic-host ; Name of
host template to use
    host_name server02
    alias Server 02
    address 192.168.1.133;
}

# Define a service to check the disk space of the root partition
# on the local machine. Warning if < 20% free, critical if
# < 10% free space on partition.

define service{
    use generic-service ;
Name of service template to use
    host_name server02
    service_description DNS
    check_command check_dns!192.168.1.133
}
```

Figura 11.2: Configurando archivo de configuración de Nagios.

Reiniciaremos el servicio nagios3 en el servidor con:

```
:~$sudo service nagios3 restart
```

Vamos a añadir una definición de servicio para la comprobación de MySQL, editando el archivo /etc/nagios3/conf.d/services_nagios2.cfg. El archivo, una vez configurado debe quedar así:

The screenshot shows a GIMP image editor window titled "server02.cfg (/etc/nagios3/conf.d) - gedit". The window contains a configuration file for Nagios. The code is as follows:

```
# Define a service to check the load on the local machine.

define service{
    use generic-service ;
    Name of service template to use
    host_name localhost
    service_description Current Load
    check_command check_load!5.0!4.0!
3.0!10.0!6.0!4.0
}

# NRPE disk check.
define service {
    use generic-service
    host_name server02
    service_description nrpe-disk
    check_command check_nrpe_1arg!check_all_disks!
192.168.1.133
}
```

The bottom status bar shows "Plain Text" and "Tab Width: 8". The cursor is positioned at the end of the last line of the configuration.

Figura 11.3: Añadiendo una definición de servicio para comprobación de MySQL.

Añadiremos también un grupo de anfitriones mysql-servers. Hay que editar el archivo /etc/nagios3/conf.d/hostgroups_nagios2.cfg. Una vez el archivo ha sido configurado, debe quedar así:

```

hostgroups_nagios2.cfg (/etc/nagios3/conf.d) - gedit
File Edit View Search Tools Documents Help
hostgroups_nagios2.cfg X
# A list of your web servers
define hostgroup {
    hostgroup_name  http-servers
        alias          HTTP servers
        members         localhost
}
# A list of your ssh-accessible servers
define hostgroup {
    hostgroup_name  ssh-servers
        alias          SSH servers
        members         localhost
}
# MySQL hostgroup.
define hostgroup {
    hostgroup_name  mysql-servers
        alias          MySQL servers
        members         localhost, server02
}
Plain Text ▾ Tab Width: 8 ▾ Ln 30, Col 1 INS

```

Figura 11.4: Añadiendo grupo de anfitriones de MySQL.

Dado que la comprobacion de Nagios necesita ingresar en MySQL, necesitamos un usuario que validar. Para ello en mysql crearemos un usuario nagios. Ejecutamos el comando:

```
:~$mysql -u root -p -e "create user nagios identified by '[ tu_password ] ';"
```

```

carlos@ubuntu:~ carlos@ubuntu:~ 81x29
carlos@ubuntu:~$ mysql -u root -p -e "create user nagios identified by 'practicas',ISE';"
Enter password:
carlos@ubuntu:~$ 

```

Figura 11.5: Creando el usuario nagios para MySQL.

Ahora, de nuevo, reiniciamos el servicio.

Tenemos que configurar el servidor nrpe para comprobar el espacio de disco en server02. Ésto se añade al archivo /etc/nagios3/conf.d/server02.cfg anteriormente creado. Dicho archivo debe quedar así tras el cambio:

```

server02.cfg (/etc/nagios3/conf.d) - gedit
File Edit View Search Tools Documents Help
Open Save Undo | Generic Service | Cut Copy Paste Find | server02.cfg X
# Define a service to check the load on the local machine.

define service{
    use generic-service ;
    Name of service template to use
    host_name localhost
    service_description Current Load
    check_command check_load!5.0!4.0!
3.0!10.0!6.0!4.0
}

# NRPE disk check.
define service {
    use generic-service
    host_name server02
    service_description nrpe-disk
    check_command check_nrpe_1arg!check_all_disks!
192.168.1.136
}

```

Figura 11.6: Añadiendo la comprobación de espacio en disco.

Así mismo, deberemos editar el archivo /etc/nagios/nrpe.cfg (Esto se hace en server02), cambiando algunas cosas. El archivo debe contener las dos líneas siguientes, de nuevo la IP que se ve es la de server02:

```

nrpe.cfg X
command[check_users]=/usr/lib/nagios/plugins/check_users -w 5 -c 10
command[check_load]=/usr/lib/nagios/plugins/check_load -w 15,10,5 -c 30,25,20
command[check_hd1]=/usr/lib/nagios/plugins/check_disk -w 20% -c 10% -p /dev/hd1
command[check_zombie_procs]=/usr/lib/nagios/plugins/check_procs -w 5 -c 10 -s Z
command[check_total_procs]=/usr/lib/nagios/plugins/check_procs -w 150 -c 200
command[check_all_disks]=/usr/lib/nagios/plugins/check_disk -w 20% -c 10% -e

# The following examples allow user-supplied arguments and can
# only be used if the NRPE daemon was compiled with support for
# command arguments *AND* the dont_blame_nrpe directive in this
# config file is set to "1". This poses a potential security risk,
# so don't use them unless you know what you're doing.

# NOTE: This option is ignored if NRPE is running under either inetd
# or xinetd
allowed_hosts=192.168.1.133

```

(a) Cambiando allowed_hosts.

(b) Añadiendo check_all_disks.

Figura 11.7: Editanto /etc/nagios/nrpe.cfg.

Ahora reiniciaremos los servicios siguientes (El primero también se debe reiniciar en server02):

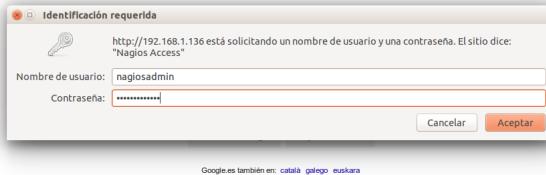
```

:~$sudo service nagios-nrpe-server restart
:~$sudo service nagios3 restart

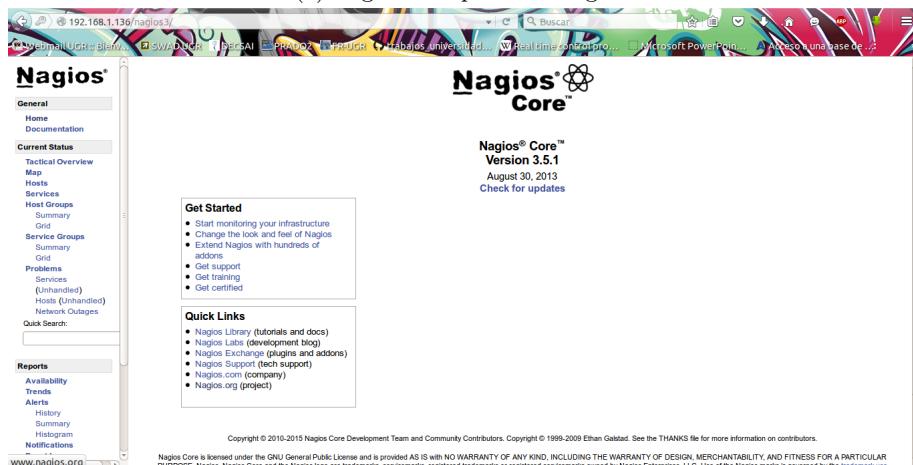
```

Ahora, desde la máquina real, accedemos a [http://\[ip_servidor\]/nagios3](http://[ip_servidor]/nagios3) (¡¡cuidado!!, ésta IP no es la de server02, sino la del servidor (máquina virtual), pero la estamos

metiendo a través de server02 (máquina real) porque hemos configurado para acceder en remoto. Fíjate que el servidor no tiene interfaz gráfica). Ingresando el nombre de usuario nagiosadmin y la contraseña que definimos al principio.



(a) Login en el panel de nagios.



(b) Panel principal de nagios.

Figura 11.8: Accediendo remotamente a nagios.

Una vez superado el proceso de instalación, vamos a monitorizar, por ejemplo las averías (breakdowns) del servidor (server01):

Host State Breakdowns:

State	Type / Reason	Time	% Total Time	% Known Time
UP	Unscheduled	0d 0h 0m 0s	0.000%	0.000%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	0d 0h 0m 0s	0.000%	0.000%
DOWN	Unscheduled	0d 0h 0m 0s	0.000%	0.000%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	0d 0h 0m 0s	0.000%	0.000%
UNREACHABLE	Unscheduled	0d 0h 0m 0s	0.000%	0.000%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	0d 0h 0m 0s	0.000%	0.000%
Undetermined	Nagios Not Running	0d 0h 0m 0s	0.000%	
	Insufficient Data	1d 0h 0m 0s	100.000%	
	Total	1d 0h 0m 0s	100.000%	
All	Total	1d 0h 0m 0s	100.000%	100.000%

State Breakdowns For Host Services:

Service	% Time OK	% Time Warning	% Time Unknown	% Time Critical	% Time Undetermined
Current Load	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	100.000%
Current Users	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	100.000%
Disk Space	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	100.000%
HTTP	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	100.000%
MySQL	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	7.440% (100.000%)	92.560%
SSH	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	100.000%
Total Processes	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	100.000%
Average	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	1.063% (14.286%)	98.937%

Host Log Entries:

[View full log entries]

Event Start Time Event End Time Event Duration Event/State Type Event/State Information

Figura 11.9: Monitorizando averías.

Tiene mucho que ver que se acabe de instalar nagios, pero como podemos observar, sólo hubo averías en MySQL, de tipo crítico. Si clicamos en MySQL podemos ver el log de lo que ha ocurrido, un acceso denegado:

Service State Breakdowns:					
State	Type / Reason	Time	% Total Time	% Known Time	
OK	Unscheduled	0d 0h 0m 0s	0.000%	0.000%	
	Scheduled	0d 0h 0m 0s	0.000%	0.000%	
	Total	0d 0h 0m 0s	0.000%	0.000%	
WARNING	Unscheduled	0d 0h 0m 0s	0.000%	0.000%	
	Scheduled	0d 0h 0m 0s	0.000%	0.000%	
	Total	0d 0h 0m 0s	0.000%	0.000%	
UNKNOWN	Unscheduled	0d 0h 0m 0s	0.000%	0.000%	
	Scheduled	0d 0h 0m 0s	0.000%	0.000%	
	Total	0d 0h 0m 0s	0.000%	0.000%	
CRITICAL	Unscheduled	0d 1h 47m 8s	7.440%	100.000%	
	Scheduled	0d 0h 0m 0s	0.000%	0.000%	
	Total	0d 1h 47m 8s	7.440%	100.000%	
Undetermined	Nagios Not Running	0d 0h 0m 0s	0.000%		
	Insufficient Data	0d 22h 12m 52s	92.560%		
	Total	0d 22h 12m 52s	92.560%		
All	Total	1d 0h 0m 0s	100.000%	100.000%	

Service Log Entries:					
[View full log entries]					
Event Start Time	Event End Time	Event Duration	Event/State Type	Event/State Information	
2015-11-25 14:29:53	2015-11-25 14:38:33	0d 0h 8m 40s	SERVICE CRITICAL (HARD)	Access denied for user 'nagios'@'localhost' (using password: YES)	

Figura 11.10: Log de un problema en MySQL.

Para monitorizar de una forma general, podemos hacer clic desde la página principal de Nagios en Tactical Overview y, en la parte superior derecha, monitorizar tiempos de ejecución y latencia:

Program-Wide Performance Information																																																					
Services Actively Checked:		<table border="1"> <thead> <tr> <th>Time Frame</th><th>Services Checked</th><th>Metric</th><th>Min.</th><th>Max.</th></tr> </thead> <tbody> <tr> <td><= 1 minute:</td><td>2 (20.0%)</td><td>Check Execution Time:</td><td>0.00 sec</td><td>10.01 sec</td></tr> <tr> <td><= 5 minutes:</td><td>10 (100.0%)</td><td>Check Latency:</td><td>0.01 sec</td><td>0.24 sec</td></tr> <tr> <td><= 15 minutes:</td><td>10 (100.0%)</td><td>Percent State Change:</td><td>0.00%</td><td>3.75%</td></tr> <tr> <td><= 1 hour:</td><td>10 (100.0%)</td><td></td><td></td><td>0.38%</td></tr> <tr> <td>Since program start:</td><td>10 (100.0%)</td><td></td><td></td><td></td></tr> </tbody> </table>				Time Frame	Services Checked	Metric	Min.	Max.	<= 1 minute:	2 (20.0%)	Check Execution Time:	0.00 sec	10.01 sec	<= 5 minutes:	10 (100.0%)	Check Latency:	0.01 sec	0.24 sec	<= 15 minutes:	10 (100.0%)	Percent State Change:	0.00%	3.75%	<= 1 hour:	10 (100.0%)			0.38%	Since program start:	10 (100.0%)																					
Time Frame	Services Checked	Metric	Min.	Max.																																																	
<= 1 minute:	2 (20.0%)	Check Execution Time:	0.00 sec	10.01 sec																																																	
<= 5 minutes:	10 (100.0%)	Check Latency:	0.01 sec	0.24 sec																																																	
<= 15 minutes:	10 (100.0%)	Percent State Change:	0.00%	3.75%																																																	
<= 1 hour:	10 (100.0%)			0.38%																																																	
Since program start:	10 (100.0%)																																																				
Services Passively Checked:		<table border="1"> <thead> <tr> <th>Time Frame</th><th>Services Checked</th><th>Metric</th><th>Min.</th><th>Max.</th></tr> </thead> <tbody> <tr> <td><= 1 minute:</td><td>0 (0.0%)</td><td>Percent State Change:</td><td>0.00%</td><td>0.00%</td></tr> <tr> <td><= 5 minutes:</td><td>0 (0.0%)</td><td></td><td></td><td></td></tr> <tr> <td><= 15 minutes:</td><td>0 (0.0%)</td><td></td><td></td><td></td></tr> <tr> <td><= 1 hour:</td><td>0 (0.0%)</td><td></td><td></td><td></td></tr> <tr> <td>Since program start:</td><td>0 (0.0%)</td><td></td><td></td><td></td></tr> </tbody> </table>				Time Frame	Services Checked	Metric	Min.	Max.	<= 1 minute:	0 (0.0%)	Percent State Change:	0.00%	0.00%	<= 5 minutes:	0 (0.0%)				<= 15 minutes:	0 (0.0%)				<= 1 hour:	0 (0.0%)				Since program start:	0 (0.0%)																					
Time Frame	Services Checked	Metric	Min.	Max.																																																	
<= 1 minute:	0 (0.0%)	Percent State Change:	0.00%	0.00%																																																	
<= 5 minutes:	0 (0.0%)																																																				
<= 15 minutes:	0 (0.0%)																																																				
<= 1 hour:	0 (0.0%)																																																				
Since program start:	0 (0.0%)																																																				
Hosts Actively Checked:		<table border="1"> <thead> <tr> <th>Time Frame</th><th>Hosts Checked</th><th>Metric</th><th>Min.</th><th>Max.</th></tr> </thead> <tbody> <tr> <td><= 1 minute:</td><td>1 (50.0%)</td><td>Check Execution Time:</td><td>0.00 sec</td><td>0.01 sec</td></tr> <tr> <td><= 5 minutes:</td><td>2 (100.0%)</td><td>Check Latency:</td><td>0.07 sec</td><td>0.14 sec</td></tr> <tr> <td><= 15 minutes:</td><td>2 (100.0%)</td><td>Percent State Change:</td><td>0.00%</td><td>0.00%</td></tr> <tr> <td><= 1 hour:</td><td>2 (100.0%)</td><td></td><td></td><td></td></tr> <tr> <td>Since program start:</td><td>2 (100.0%)</td><td></td><td></td><td></td></tr> </tbody> </table>				Time Frame	Hosts Checked	Metric	Min.	Max.	<= 1 minute:	1 (50.0%)	Check Execution Time:	0.00 sec	0.01 sec	<= 5 minutes:	2 (100.0%)	Check Latency:	0.07 sec	0.14 sec	<= 15 minutes:	2 (100.0%)	Percent State Change:	0.00%	0.00%	<= 1 hour:	2 (100.0%)				Since program start:	2 (100.0%)																					
Time Frame	Hosts Checked	Metric	Min.	Max.																																																	
<= 1 minute:	1 (50.0%)	Check Execution Time:	0.00 sec	0.01 sec																																																	
<= 5 minutes:	2 (100.0%)	Check Latency:	0.07 sec	0.14 sec																																																	
<= 15 minutes:	2 (100.0%)	Percent State Change:	0.00%	0.00%																																																	
<= 1 hour:	2 (100.0%)																																																				
Since program start:	2 (100.0%)																																																				
Hosts Passively Checked:		<table border="1"> <thead> <tr> <th>Time Frame</th><th>Hosts Checked</th><th>Metric</th><th>Min.</th><th>Max.</th></tr> </thead> <tbody> <tr> <td><= 1 minute:</td><td>0 (0.0%)</td><td>Percent State Change:</td><td>0.00%</td><td>0.00%</td></tr> <tr> <td><= 5 minutes:</td><td>0 (0.0%)</td><td></td><td></td><td></td></tr> <tr> <td><= 15 minutes:</td><td>0 (0.0%)</td><td></td><td></td><td></td></tr> <tr> <td><= 1 hour:</td><td>0 (0.0%)</td><td></td><td></td><td></td></tr> <tr> <td>Since program start:</td><td>0 (0.0%)</td><td></td><td></td><td></td></tr> </tbody> </table>				Time Frame	Hosts Checked	Metric	Min.	Max.	<= 1 minute:	0 (0.0%)	Percent State Change:	0.00%	0.00%	<= 5 minutes:	0 (0.0%)				<= 15 minutes:	0 (0.0%)				<= 1 hour:	0 (0.0%)				Since program start:	0 (0.0%)																					
Time Frame	Hosts Checked	Metric	Min.	Max.																																																	
<= 1 minute:	0 (0.0%)	Percent State Change:	0.00%	0.00%																																																	
<= 5 minutes:	0 (0.0%)																																																				
<= 15 minutes:	0 (0.0%)																																																				
<= 1 hour:	0 (0.0%)																																																				
Since program start:	0 (0.0%)																																																				
Check Statistics:		<table border="1"> <thead> <tr> <th>Type</th><th>Last 1 Min</th><th>Last 5 Min</th><th>Last 15 Min</th></tr> </thead> <tbody> <tr> <td>Active Scheduled Host Checks</td><td>1</td><td>2</td><td>6</td></tr> <tr> <td>Active On-Demand Host Checks</td><td>1</td><td>4</td><td>12</td></tr> <tr> <td>Parallel Host Checks</td><td>1</td><td>2</td><td>6</td></tr> <tr> <td>Serial Host Checks</td><td>0</td><td>0</td><td>0</td></tr> <tr> <td>Cached Host Checks</td><td>1</td><td>4</td><td>12</td></tr> <tr> <td>Passive Host Checks</td><td>0</td><td>0</td><td>0</td></tr> <tr> <td>Active Scheduled Service Checks</td><td>2</td><td>10</td><td>30</td></tr> <tr> <td>Active On-Demand Service Checks</td><td>0</td><td>0</td><td>0</td></tr> <tr> <td>Cached Service Checks</td><td>0</td><td>0</td><td>0</td></tr> <tr> <td>Passive Service Checks</td><td>0</td><td>0</td><td>0</td></tr> <tr> <td>External Commands</td><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>				Type	Last 1 Min	Last 5 Min	Last 15 Min	Active Scheduled Host Checks	1	2	6	Active On-Demand Host Checks	1	4	12	Parallel Host Checks	1	2	6	Serial Host Checks	0	0	0	Cached Host Checks	1	4	12	Passive Host Checks	0	0	0	Active Scheduled Service Checks	2	10	30	Active On-Demand Service Checks	0	0	0	Cached Service Checks	0	0	0	Passive Service Checks	0	0	0	External Commands	0	0	0
Type	Last 1 Min	Last 5 Min	Last 15 Min																																																		
Active Scheduled Host Checks	1	2	6																																																		
Active On-Demand Host Checks	1	4	12																																																		
Parallel Host Checks	1	2	6																																																		
Serial Host Checks	0	0	0																																																		
Cached Host Checks	1	4	12																																																		
Passive Host Checks	0	0	0																																																		
Active Scheduled Service Checks	2	10	30																																																		
Active On-Demand Service Checks	0	0	0																																																		
Cached Service Checks	0	0	0																																																		
Passive Service Checks	0	0	0																																																		
External Commands	0	0	0																																																		
Buffer Usage:		<table border="1"> <thead> <tr> <th>Type</th><th>In Use</th><th>Max Used</th><th>Total Available</th></tr> </thead> <tbody> <tr> <td>External Commands</td><td>0</td><td>0</td><td>4096</td></tr> </tbody> </table>				Type	In Use	Max Used	Total Available	External Commands	0	0	4096																																								
Type	In Use	Max Used	Total Available																																																		
External Commands	0	0	4096																																																		

Figura 11.11: Monitorizando tiempos de ejecución del servidor.

Donde podemos observar todos los servicios que han sido comprobados activa y pasivamente. Además, Nagios nos ofrece algunas estadísticas al respecto que podríamos usar para varios cometidos.

(Fuentes: [5])

Referencias

- [1] Debian Administration. Using strace to debug application errors. https://www.debian-administration.org/article/352/Using_strace_to_debug_application_errors.
- [2] CodeJobs. Cómo crear una tabla en mysql. <https://www.codejobs.biz/es/blog/2012/08/20/como-crear-una-tabla-en-mysql>.
- [3] Blog desde linux. Cron & crontab, explicados. <http://blog.desdelinux.net/cron-crontab-explicados/>.
- [4] Blog desde linux. Tip: comando dd con barra de progreso. <http://blog.desdelinux.net/tip-comando-dd-con-barra-de-progreso/>.
- [5] Ubuntu documentation. Nagios. <https://help.ubuntu.com/lts/serverguide/nagios.html>.
- [6] Erjaimer. Auditar un ejecutable en gnu/linux (i). <https://erjaimer.wordpress.com/tag/forense-de-aplicaciones-gnulinux/>.
- [7] Kriptópolis. Usando raid en casa (2): Creación y gestión de un raid software. <http://www.kriptopolis.com/raid-2>.
- [8] Ubuntu Manpage. sg_modes. http://manpages.ubuntu.com/manpages/natty/man8/sg_modes.8.html.
- [9] Munin. Demo de munin. <http://demo.munin-monitoring.org/>.
- [10] MySQL. Show profile syntax. <http://dev.mysql.com/doc/refman/5.0/en/show-profile.html>.
- [11] Linux RAID. Detecting, querying and testing. https://raid.wiki.kernel.org/index.php/Detecting,_querying_and_testing.
- [12] RedHat. How to use yum history to roll back an update in red hat enterprise linux 6, 7? <https://access.redhat.com/solutions/64069>.
- [13] rm_rf. Visualizar histórico de instalaciones por apt. <http://rm-rf.es/visualizar-historico-de-instalaciones-por-apt/>.
- [14] The source is with you. How to monitor hardware sensors in ubuntu #15. https://www.youtube.com/watch?v=_BMFdTHYhoA.
- [15] Microsoft Technet. Crear un conjunto de recopiladores de datos desde el monitor de rendimiento. <https://technet.microsoft.com/es-es/library/cc722148.aspx>.
- [16] Microsoft Technet. Perfmon. <https://technet.microsoft.com/en-us/library/cc731067%28v=ws.10%29.aspx>.

- [17] Microsoft Technet. Windows performance monitor. <https://technet.microsoft.com/en-us/library/cc749249.aspx?ppud=4>.
- [18] Ask Ubuntu. How do i find out what filesystem my partitions are using? <http://askubuntu.com/questions/309047/how-do-i-find-out-what-filesystem-my-partitions-are-using>.
- [19] Guia Ubuntu. Crear una software raid. http://www.guia-ubuntu.com/index.php?title=Crear_una_Software_RAID#Formatear_la_RAID.
- [20] Ubuntu-es. Añadir al nombre de un fichero la fecha y hora ctual. <http://www.ubuntu-es.org/node/111928#.Vktvf5dVK1E>.
- [21] Wikipedia. dmesg. <https://es.wikipedia.org/wiki/Dmesg>.
- [22] Wpitchoune. Psensor readme. http://wpitchoune.net/psensor/doc/README.html#_ati_amd_gpu_support.
- [23] Estrellate y arde. Logs en linux. <http://www.estrellateyarde.org/logs-en-linux>.