

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

---

## Memoria Práctica 3

---

Antonio Javier Cabrera Gutiérrez

1 de diciembre de 2015

## Índice

1. Cuestión 1	5
1.1. ¿Qué archivo le permite ver que 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. Indique que comando ha utilizado para realizarlo así como capturas de pantalla del proceso de reconstrucción del RAID	5
3. ¿Qué archivo ha de modificar para programar una tarea? Escriba la linea 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
4. Pruebe a ejecutar el comando, conectar un dispositivo USB y vuelva a ejecutar el comando. Copie y pegue la salida del comando. (considere usar dmsg   tail). Comente que observa en la información mostrada.	8
5. Ejecute el monitor de "System Performancez muestre el resultado. Incluya capturas de pantalla comentando la información que aparece	8
6. Cree un recopilador de datos definido por el usuario(modos 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	11
7. Instale alguno de los monitores comentados arriba en su maquina y pruebe a ejecutarlos (tenga en cuenta que si lo hace en la maquina virtual los resultados pueden no ser realistas). Alternativamente, busque otros monitores para hardware comerciales o de código abierto para Windows y Linux	14
8. Visite la web del proyecto y acceda a la demo que proporcionan donde se muestra como monitorizan un servidor. Monitoree varios parámetros y haga capturas de pantalla de lo que esta mostrando comentando que observa.	15
9. Haga lo mismo que con Munin	20
10. Instale el monitor y muestre y comente algunas capturas de pantalla	23
11. Escriba un breve resumen sobre alguno de los artículos donde se muestra el uso de strace o busque otro y comentalo	26

12.Desarrolle una pagina en C o en C++ y analice su comportamiento usando valdgrind.	26
13.Escriba un script en python y analice su comportamiento usando el profiler presentado	27
14.Acceda a la consola de mysql y muestre el resultado de mostrar el profile de una consulta	28
15.Al igual que ha realizado el profiling con MySQL, realice lo mismo con MongoDB y compare los resultados (use la misma informacion y la misma consulta)	30

## Índice de figuras

2.1. Discos del Raid . . . . .	6
2.2. Raid con un disco eliminado . . . . .	6
2.3. Monotorizacion sin el disco . . . . .	6
2.4. Reconstrucción del raid . . . . .	7
2.5. Raid reconstruido . . . . .	7
4.1. Salida de la orden dmsg . . . . .	8
5.1. Informe System Performance 1 . . . . .	9
5.2. Informe System Performance 2 . . . . .	10
6.1. Creacion del monitor . . . . .	11
6.2. Configuracion avanzada . . . . .	11
6.3. Selecccion de registro de datos . . . . .	12
6.4. Eleccion de lo que queremos monitorizar . . . . .	12
6.5. Selecccion de tiempo de muestreo . . . . .	13
6.6. Ruta donde guardar los archivos . . . . .	13
7.1. Añadiendo chips a analizar con lmsensors . . . . .	14
7.2. Ejecutando lmsensors . . . . .	15
8.1. Interfaces de red . . . . .	16
8.2. Tanto por ciento de uso de la CPU . . . . .	16
8.3. Acesos y procesos de apache . . . . .	17
8.4. Estadísticas del disco . . . . .	18
8.5. Procesos . . . . .	18
8.6. Carga media del sistema . . . . .	19
8.7. Uso de la memoria . . . . .	19
9.1. Panoramica del host . . . . .	20
9.2. Medidas de cpu . . . . .	21
9.3. Estadisticas de disco . . . . .	21
9.4. Carga media del sistema . . . . .	22
9.5. Estadisticas de red y de procesos . . . . .	22
9.6. Estadisticas de la memoria . . . . .	23

10.1. Pagina inicial de Awstats . . . . .	24
10.2. Histórico de días de la semana y de horas . . . . .	25
10.3. Otros datos de awstat . . . . .	25
12.1. Ejecucion de valgrind . . . . .	27
13.1. Ejecucion del profiler . . . . .	28
14.1. Creacion de la tabla . . . . .	28
14.2. Insercion de datos 1 . . . . .	28
14.3. Insercion de datos 2 . . . . .	28
14.4. Insercion de datos 3 . . . . .	28
14.5. Activacion del profilin y consulta sql . . . . .	29
14.6. Tiempo de la consulta . . . . .	29
14.7. Visualizacion de los tiempos medidos de la consulta . . . . .	29
15.1. Creacion de tablas . . . . .	30
15.2. Importacion del archivo json . . . . .	30
15.3. Consulta . . . . .	30
15.4. Profiling MongoDB . . . . .	31

## 1. Cuestión 1

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

El archivo `/var/log/apt/history.log`

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

Linux coje los ficheros `.log` y periódicamente si el archivo es muy grande los comprime en `gz`, esto se llama rotación de logs, los datos mas antiguos los pone en otros ficheros para que el primero no sea muy grande. El numero 1 o 2 es el orden en el que lo ha comprimido siendo el valor menor el ultimo en comprimir. Linux gestiona esto a través del demonio `logrotate` que esta configurado en este caso para `apt` de la siguiente forma en mi ordenador. Para obtener el archivo vamos a `/etc/logrotate.d/apt`

```
/var/log/apt/term.log {
    rotate 12
    monthly
    compress
    missingok
    notifempty
}
```

```
/var/log/apt/history.log {
    rotate 12
    monthly
    compress
    missingok
    notifempty
}
```

`rotate 12` quiere decir que se mantendrán los últimos 12 ficheros rotados. es decir el `.1.gz` hasta el `.12.gz` o sea que rotara 12 veces. `monthly` quiere decir que los archivos se rotaran cada mes. `compress` quiere decir que se comprime el archivo rotado. `missingok` quiere decir que si el fichero `log` no existe no devuelve error. <sup>1</sup>

## 2. Indique que comando ha utilizado para realizarlo así como capturas de pantalla del proceso de reconstrucción del RAID

Vemos que están los dos discos con la orden `lsblk`

---

<sup>1</sup><http://helloit.es/2012/08/rotar-archivos-de-log-con-logrotate/>

```

antonio@ubuntu:~$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                  8:0      0   10G  0 disk
├─sda1                              8:1      0   10G  0 part
│   └─md0                          9:0      0   10G  0 raid1
│       ├─HDS-arranq (dm-0)        252:0     0  200M  0 lvm  /boot
│       ├─HDS-home (dm-1)         252:1     0  500M  0 lvm
│       │   └─HDS-home_crypt (dm-6) 252:6     0  498M  0 crypt /home
│       ├─HDS-raiz (dm-2)         252:2     0    6G  0 lvm
│       │   └─HDS-raiz_crypt (dm-4) 252:4     0    6G  0 crypt /
│       ├─HDS-swap (dm-3)         252:3     0 1000M  0 lvm
│       │   └─HDS-swap_crypt (dm-5) 252:5     0  998M  0 crypt [SWAP]
└─sdb                                8:16     0   10G  0 disk
    ├─sdb1                          8:17     0   10G  0 part
    │   └─md0                      9:0      0   10G  0 raid1
    │       ├─HDS-arranq (dm-0)    252:0     0  200M  0 lvm  /boot
    │       ├─HDS-home (dm-1)     252:1     0  500M  0 lvm
    │       │   └─HDS-home_crypt (dm-6) 252:6     0  498M  0 crypt /home
    │       ├─HDS-raiz (dm-2)     252:2     0    6G  0 lvm
    │       │   └─HDS-raiz_crypt (dm-4) 252:4     0    6G  0 crypt /
    │       ├─HDS-swap (dm-3)     252:3     0 1000M  0 lvm
    │       │   └─HDS-swap_crypt (dm-5) 252:5     0  998M  0 crypt [SWAP]
└─sr0                               11:0     1 1024M  0 rom

```

Figura 2.1: Discos del Raid

Primero debemos de marcar el disco como fallido con `mdadm /dev/md0 -f /dev/sda1`  
 Una vez marcado borramos el disco con `mdadm /dev/md0 -r /dev/sda1` Miramos otra vez con la orden `lsblk` y vemos que nos sale un disco.

```

antonio@ubuntu:~$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                  8:0      0   10G  0 disk
├─sda1                              8:1      0   10G  0 part
└─sdb                                8:16     0   10G  0 disk
    ├─sdb1                          8:17     0   10G  0 part
    │   └─md0                      9:0      0   10G  0 raid1
    │       ├─HDS-arranq (dm-0)    252:0     0  200M  0 lvm  /boot
    │       ├─HDS-home (dm-1)     252:1     0  500M  0 lvm
    │       │   └─HDS-home_crypt (dm-6) 252:6     0  498M  0 crypt /home
    │       ├─HDS-raiz (dm-2)     252:2     0    6G  0 lvm
    │       │   └─HDS-raiz_crypt (dm-4) 252:4     0    6G  0 crypt /
    │       ├─HDS-swap (dm-3)     252:3     0 1000M  0 lvm
    │       │   └─HDS-swap_crypt (dm-5) 252:5     0  998M  0 crypt [SWAP]
└─sr0                               11:0     1 1024M  0 rom

```

Figura 2.2: Raid con un disco eliminado

Miramos el estado del raid con `watch -n2 cat /proc/mdstat`

```

Cada 2.0s: cat /proc/mdstat                               Fri Nov 20 15:00:30 2015
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sdb1[1]
      10475392 blocks super 1.2 [2/1] [_U]
unused devices: <none>

```

Figura 2.3: Monotorizacion sin el disco

Ahora cogemos y añadimos el disco con la orden `mdadm /dev/md0 -a /dev/sda1`

```

Cada 2,0s: cat /proc/mdstat                               Fri Nov 20 14:54:36 2015

Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sda1[2] sdb1[1]
      10475392 blocks super 1.2 [2/1] [_U]
      [====>.....] recovery = 33.3% (3498688/10475392) finish=3.7min
      speed=30686K/sec
unused devices: <none>

```

Figura 2.4: Reconstrucción del raid

Una vez acabado el proceso de reconstrucción nos sale esto con el watch

```

Cada 2,0s: cat /proc/mdstat                               Fri Nov 20 14:58:38 2015

Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sda1[2] sdb1[1]
      10475392 blocks super 1.2 [2/2] [UU]
unused devices: <none>

```

Figura 2.5: Raid reconstruido

### 3. ¿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 hay que modificar es /etc/crontab La línea ha de escribir es:<sup>2</sup>

```
0 0 * * * root cp ~/codigo ~/seguridad/$(date +%d%m%y)
```

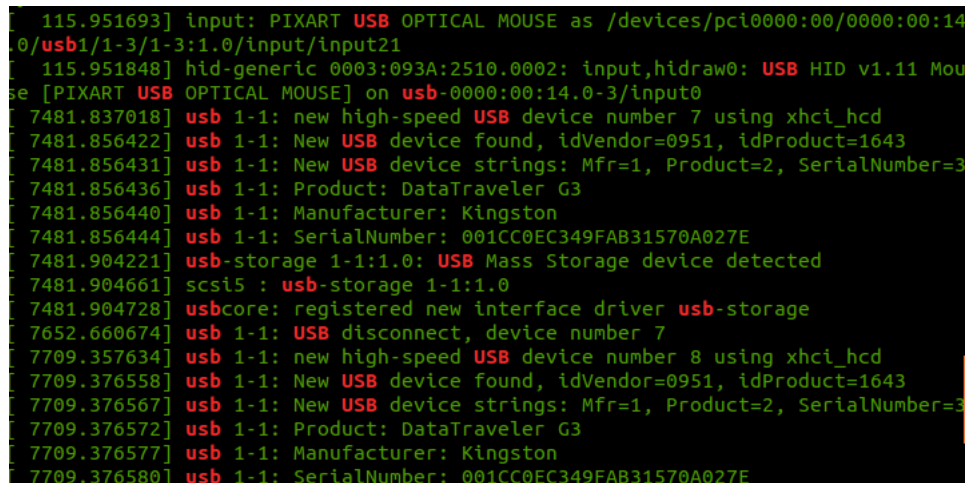
Estos campos son los siguientes:

- Corresponde al minuto en el que se va a ejecutar
- Hora en formato 24 horas en la que se va a ejecutar
- Día del mes en el que se va a ejecutar, en nuestro caso como queremos que se ejecute cada día pues ponemos un asterisco.
- Corresponde al mes
- Día de la semana que se quiere ejecutar, puede ser valores del 0 al 7, donde 0 y 7 son domingo, o poner las 3 primeras letras del día en inglés: mon, tue, etc.

<sup>2</sup><http://blog.desdelinux.net/cron-crontab-explicados/>

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

He utilizado un junto con la orden `dmsg` la orden `grep`. Entonces con `dmsg | grep -i usb` he filtrado por `usb` y se puede ver que ha reconocido el pen drive y el ratón que tengo conectado

A terminal window with a black background and green text. The output shows the detection and identification of two USB devices. The first device is a PIXART USB OPTICAL MOUSE, identified as a USB HID v1.11 Mouse. The second device is a Kingston DataTraveler G3 USB Mass Storage device. The output includes timestamps, device paths, and various identification strings like Vendor ID, Product ID, and Serial Number. The text is as follows:

```
115.951693] input: PIXART USB OPTICAL MOUSE as /devices/pci0000:00/0000:00:14.0/usb1/1-3/1-3:1.0/input/input21
115.951848] hid-generic 0003:093A:2510.0002: input,hidraw0: USB HID v1.11 Mouse [PIXART USB OPTICAL MOUSE] on usb-0000:00:14.0-3/input0
7481.837018] usb 1-1: new high-speed USB device number 7 using xhci_hcd
7481.856422] usb 1-1: New USB device found, idVendor=0951, idProduct=1643
7481.856431] usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
7481.856436] usb 1-1: Product: DataTraveler G3
7481.856440] usb 1-1: Manufacturer: Kingston
7481.856444] usb 1-1: SerialNumber: 001CC0EC349FAB31570A027E
7481.904221] usb-storage 1-1:1.0: USB Mass Storage device detected
7481.904661] scsi5 : usb-storage 1-1:1.0
7481.904728] usbcore: registered new interface driver usb-storage
7652.660674] usb 1-1: USB disconnect, device number 7
7709.357634] usb 1-1: new high-speed USB device number 8 using xhci_hcd
7709.376558] usb 1-1: New USB device found, idVendor=0951, idProduct=1643
7709.376567] usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
7709.376572] usb 1-1: Product: DataTraveler G3
7709.376577] usb 1-1: Manufacturer: Kingston
7709.376580] usb 1-1: SerialNumber: 001CC0EC349FAB31570A027E
```

Figura 4.1: Salida de la orden `dmsg`

Se puede apreciar cuando se desconecta, en la línea que pone `usb-storage 1-1:1.0: USB Mass Storage device detected` y se puede ver cuando se desconecta con `USB disconnect, device number 7`.

5. Ejecute el monitor de "System Performancez muestre el resultado. Incluya capturas de pantalla comentando la información que aparece

Al ejecutar el monitor y esperar 60 segundos el informe que nos sale es inmenso. Yo comentare algunas características de entre todas las estadísticas mostradas



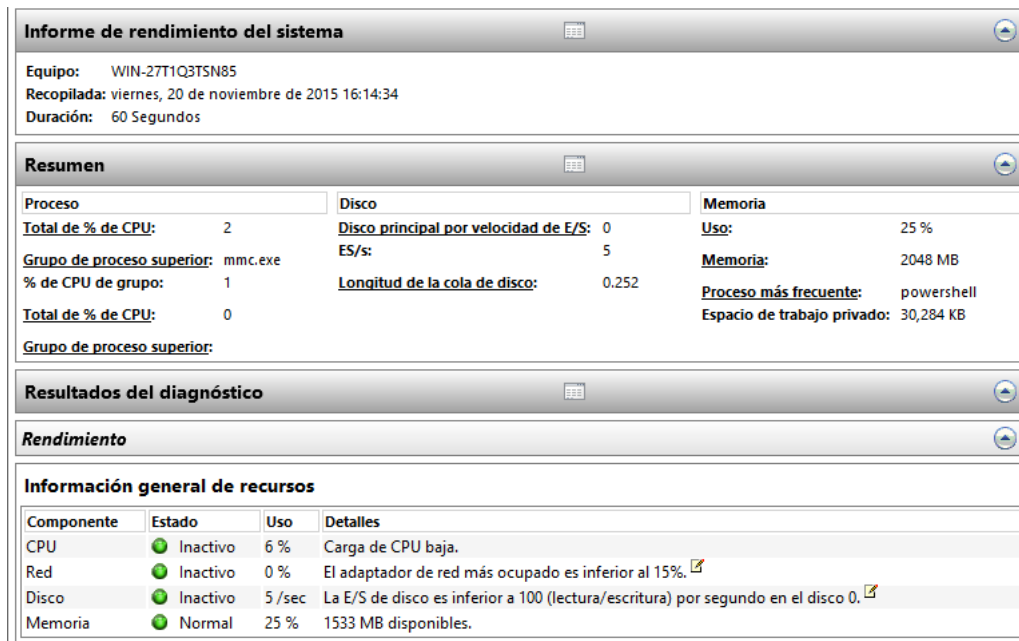


Figura 5.1: Informe System Performance 1

Como se puede ver aquí nos muestra el nombre del equipo y la fecha en la que se ha realizado la monitorización, en esta primera parte nos muestra un resumen de todo el informe: En procesador como se puede ver se ha utilizado poco , en disco se ha producido 5 accesos y en memoria se puede ver que esta usada el 25 por ciento y el tamaño de la memoria es 2048 MB. En rendimiento se puede ver que el uso de la CPU ha sido un 6 % y nos dice que la carga ha sido baja. por otro lado la tarjeta de red ha estado inactiva, el disco también, como hemos dicho antes solo ha habido 5 accesos, y la memoria su uso es algo mas alto que los otro componentes ya que el 25 por ciento esta en uso.

Proceso					
Memoria					
				Superior: 10	de 33
Proceso	Id. del proceso	Confirmar (KB)	Espacio de trabajo (KB)	Compartible (KB)	Privado (KB)
powershell	924	623,612	66,432	36,148	30,284
dwm	712	124,808	33,348	16,916	16,432
mmc	1540	348,044	49,120	33,356	15,764
svchost##3	768	122,268	24,432	14,756	9,676
svchost##2	684	59,232	15,388	5,992	9,396
explorer	1904	314,872	37,976	29,680	8,296
rundll32	2464	80,152	12,328	6,268	6,060
svchost##5	900	1,154,860	14,940	9,604	5,336
svchost##6	472	64,708	12,112	7,004	5,108
WmiPrvSE	2384	65,096	12,652	8,032	4,620

Contadores			
Memoria			
		Superior: 29	de 29
counter	Medio	Mínimo	Máximo
% de bytes confirmados en uso	12	12	12
Bytes disponibles	1,607,515,556	1,604,988,928	1,612,849,152
Bytes de caché	21,668,444	21,557,248	23,392,256
Errores de caché/s	0.917	0	24
Límite de confirmación	4,294,496,256	4,294,496,256	4,294,496,256
Bytes confirmados	520,691,376	514,576,384	524,623,872
Errores de solicitud de cero/s	353	39	1,684
Bytes de lista de páginas libres y cero	1,361,762,405	1,359,405,056	1,368,866,816
Entradas libres de la tabla de páginas del sistema	33,556,400	33,556,399	33,556,401
Duración en caché de modo de espera promedio a largo plazo (s)	14,400	14,400	14,400
Bytes de lista de páginas modificadas	15,303,730	15,077,376	15,380,130

Figura 5.2: Informe System Performance 2

En esta imagen se puede ver los procesos que había activos en el sistema durante el tiempo que ha durado la monotorizacion, con sus id's el espacio de trabajo y mas datos asociados al proceso. Con respecto a la memoria también se puede ver datos mas en detalle como los Bytes de cache, los bytes disponibles. Como se puede ver durante los 60 segundos que ha durado la monotorizacion el equipo ha estado parado y no ha trabajado. También hay que decir que la cantidad de datos ofrecidos por este informe es grandisima, solo he comentado estos pero podría haber comentados muchos mas.

6. Cree un recopilador de datos definido por el usuario(modos avanzado) que incluya tanto el contador de rendimiento como los datos de seguimiento Todos los referentes al procesador, al proceso y al servicio web.  
Intervalo de muestra 15 segundos.  
Almacene el resultado en el directorio Escritorio/logs

## Incluya las capturas de pantalla de cada paso

En la ventana de la izquierda pinchamos lo siguiente

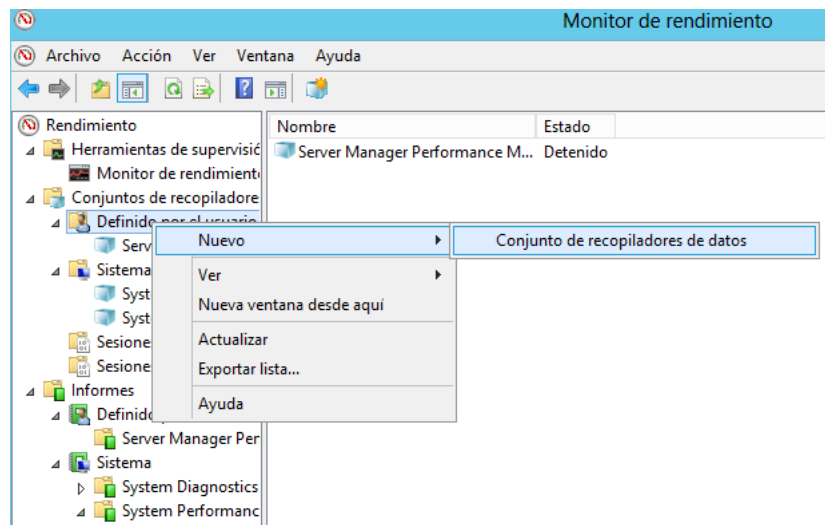


Figura 6.1: Creacion del monitor

Elegimos configuracion avanzada y le ponemos nombre

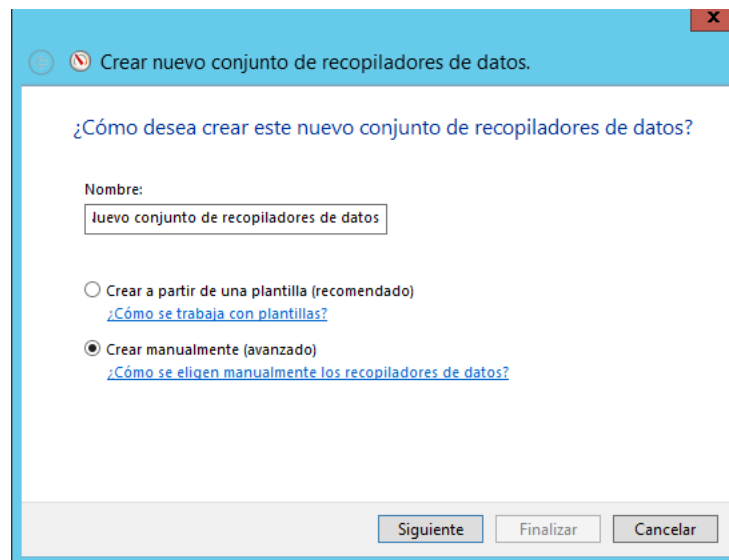


Figura 6.2: Configuracion avanzada

Seleccionamos Contador de rendimiento y datos de seguimiento de eventos

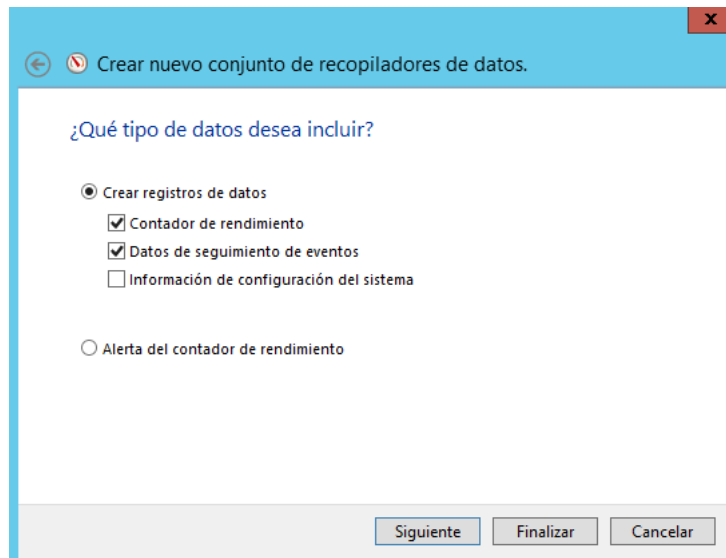


Figura 6.3: Selecccion de registro de datos

Ahora elegimos que monitorice el procesador, el proceso y el servicio web.

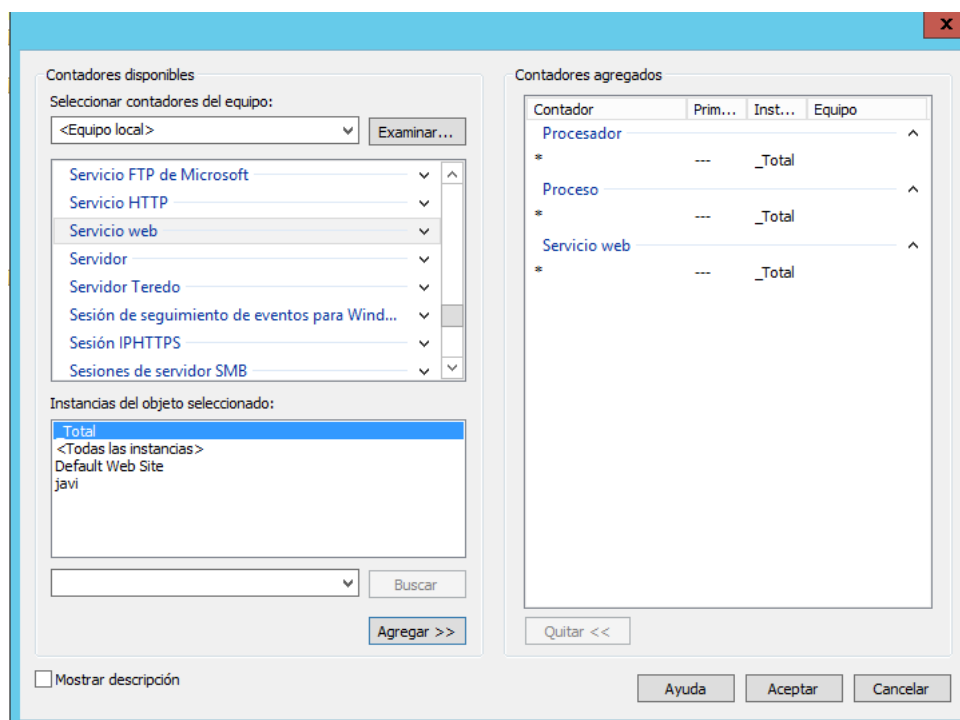


Figura 6.4: Eleccion de lo que queremos monitorizar

Ponemos el tiempo de muestreo a 15 segundos

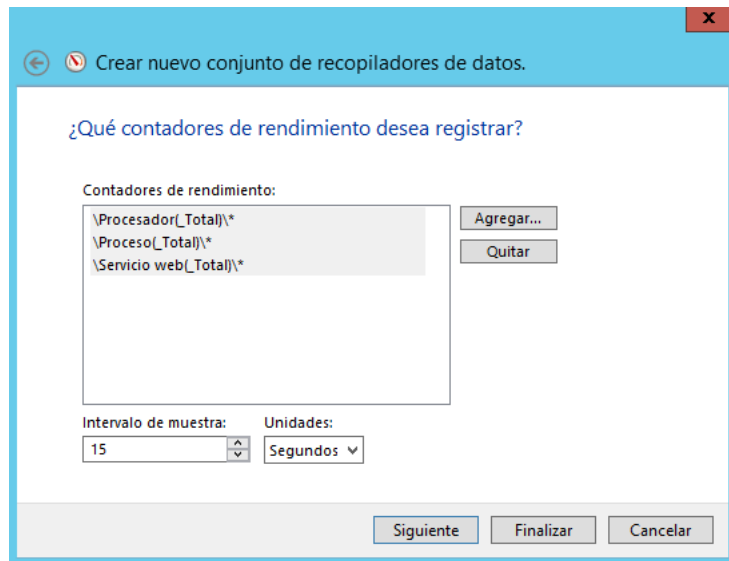


Figura 6.5: Selecccion de tiempo de muestreo

Seleccionamos la ruta donde deseamos guardar los datos

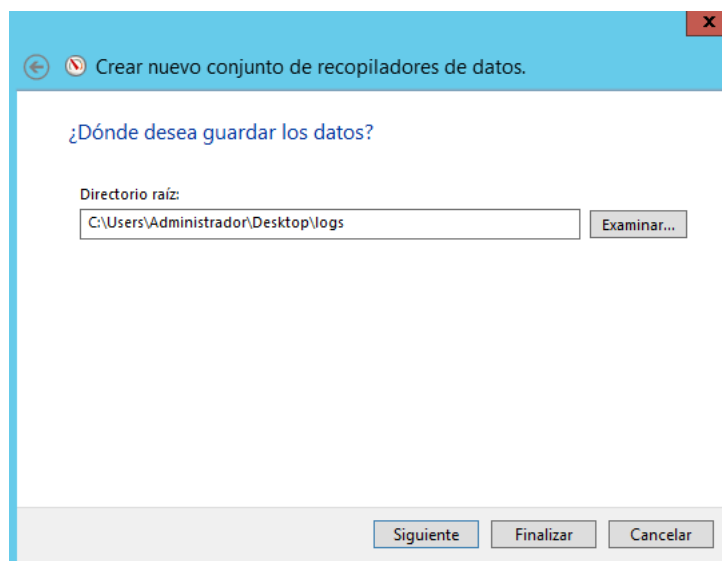


Figura 6.6: Ruta donde guardar los archivos

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

Voy a probar un monitor linux ya que el sistema operativo en el que trabajo es ubuntu, voy a probar lm-sensors. Para instalarlo hay que escribir `sudo apt-get install lm-sensors`. Usamos la orden `sensors-detect` para que pueda detectar los chips a analizar, nos realizara preguntas y le damos a yes a los que queramos analizar.

```
Probing for 'National Semiconductor LM79' at 0x290...      No
Probing for 'Winbond W83781D' at 0x290...                 No
Probing for 'Winbond W83782D' at 0x290...                 No

Lastly, we can probe the I2C/SMBus adapters for connected hardware
monitoring devices. This is the most risky part, and while it works
reasonably well on most systems, it has been reported to cause trouble
on some systems.
Do you want to probe the I2C/SMBus adapters now? (YES/no): yes
Found unknown SMBus adapter 8086:9ca2 at 0000:00:1f.3.
Sorry, no supported PCI bus adapters found.
Module i2c-dev loaded successfully.

Next adapter: i915 gmbus ssc (i2c-0)
Do you want to scan it? (yes/NO/selectively): yes

Next adapter: i915 gmbus vga (i2c-1)
Do you want to scan it? (yes/NO/selectively): yes

Next adapter: i915 gmbus panel (i2c-2)
Do you want to scan it? (yes/NO/selectively): yes

Next adapter: i915 gmbus dpc (i2c-3)
Do you want to scan it? (yes/NO/selectively): █
```

Figura 7.1: Añadiendo chips a analizar con lmsensors

Ahora al final nos preguntara "Do you want to add these lines to `/etc/modules` automatically? (yes/NO)", le damos a yes para que añada los módulos a analizar. Yo solo tengo añadido `coretemp`, que pide la temperatura de los procesadores. Al ejecutar la orden `sensors` vemos lo que nos sale:

```

usuario@usuario-X555LDB:~$ sensors
acpitz-virtual-0
Adapter: Virtual device
temp1:      +34.0°C (crit = +108.0°C)
temp2:      +27.8°C (crit = +105.0°C)
temp3:      +29.8°C (crit = +105.0°C)

coretemp-isa-0000
Adapter: ISA adapter
Physical id 0: +34.0°C (high = +105.0°C, crit = +105.0°C)
Core 0:      +33.0°C (high = +105.0°C, crit = +105.0°C)
Core 1:      +34.0°C (high = +105.0°C, crit = +105.0°C)

asus-isa-0000
Adapter: ISA adapter
temp1:      +34.0°C

nouveau-pci-0400
Adapter: PCI adapter
temp1:      N/A (high = +95.0°C, hyst = +3.0°C)
              (crit = +105.0°C, hyst = +5.0°C)
              (emerg = +135.0°C, hyst = +5.0°C)

```

Figura 7.2: Ejecutando lmsensors

Otros monitores que podemos encontrar es HWMonitor.<sup>3</sup> Este monitor permite medir la temperatura, voltaje, el consumo y la utilización de la CPU. Temperatura del disco duro, y voltaje, temperatura y utilización de la GPU. También existe otro llamado Speedfan en el que podemos ver voltajes y temperaturas de los componentes de nuestro equipo así como la frecuencia de los ventiladores y el uso de la CPU.<sup>4</sup>

## 8. Visite la web del proyecto y acceda a la demo que proporcionan donde se muestra como monitorizan un servidor. Monitoree varios parámetros y haga capturas de pantalla de lo que esta mostrando comentando que observa.

Cuando accedemos a la demo la cantidad de estadísticas y gráficas es muy grande. Comentare las que me han parecido mas relevantes

<sup>3</sup><http://www.cpuid.com/softwares/hwmonitor.html>

<sup>4</sup><http://www.almico.com/speedfan.php>

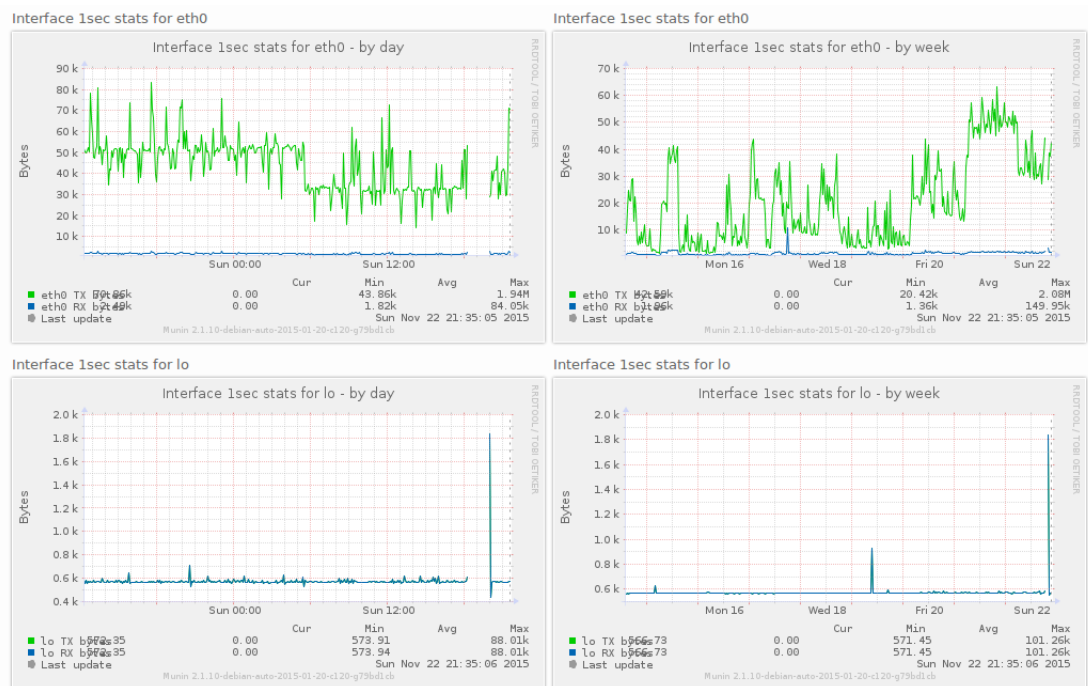


Figura 8.1: Interfaces de red

Primero comentamos la estadísticas de interfaz de red tanto la eth0 y la lo. como vemos se nos muestra tanto los kbytes de subida como los de bajada.

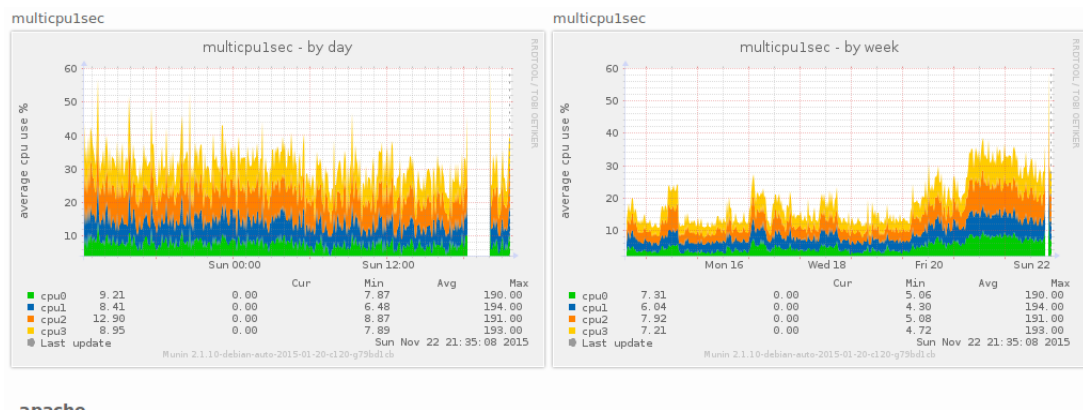


Figura 8.2: Tanto por ciento de uso de la CPU

En esta gráfica se nos muestra el tanto por ciento de uso de la CPU, como vemos tiene 4 procesadores, durante la semana los cuatro procesadores se han usado mas o menos lo mismo.



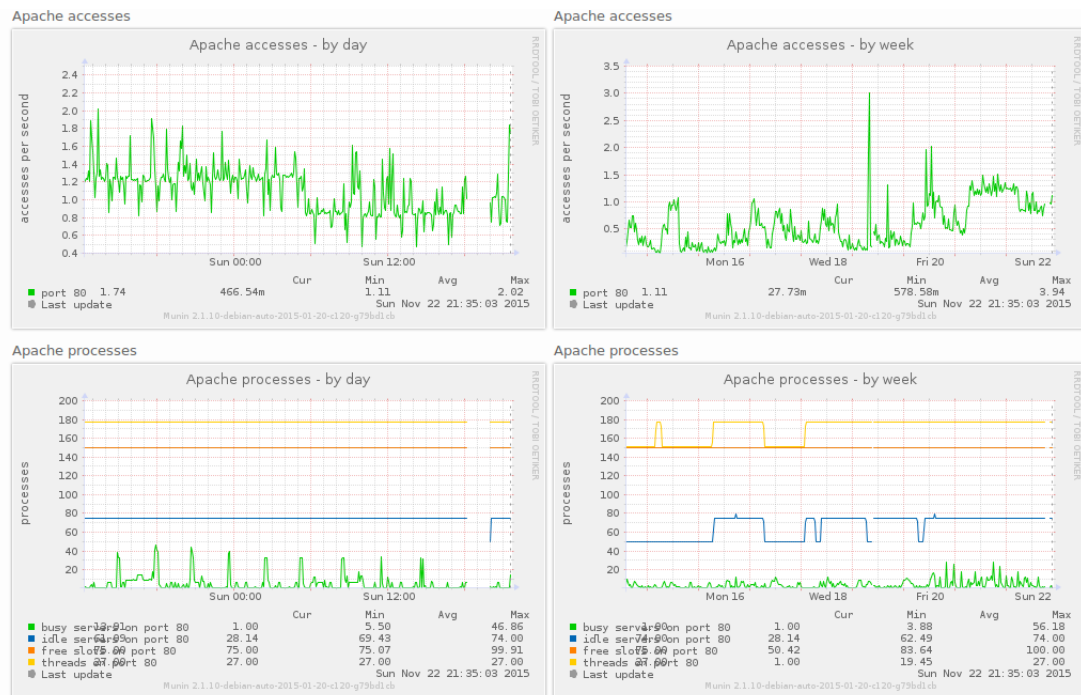


Figura 8.3: Acesos y procesos de apache

En esta imagen se puede ver en las dos primeras gráficas el trafico de datos del puerto 80, o sea apache, y las dos gráficas de abajo se puede ver los procesos ejecutados en apache.



Figura 8.4: Estadísticas del disco

Aquí podemos ver en las dos gráficas de arriba las entradas y salidas en disco, en positivo se muestran las escrituras y en negativo las lecturas. En las gráficas de abajo se muestra la latencia de disco, esto es el tiempo de acceso a disco, se puede apreciar que durante el fin de semana el tiempo de acceso a disco es mucho menor que en los días laborables de la semana.

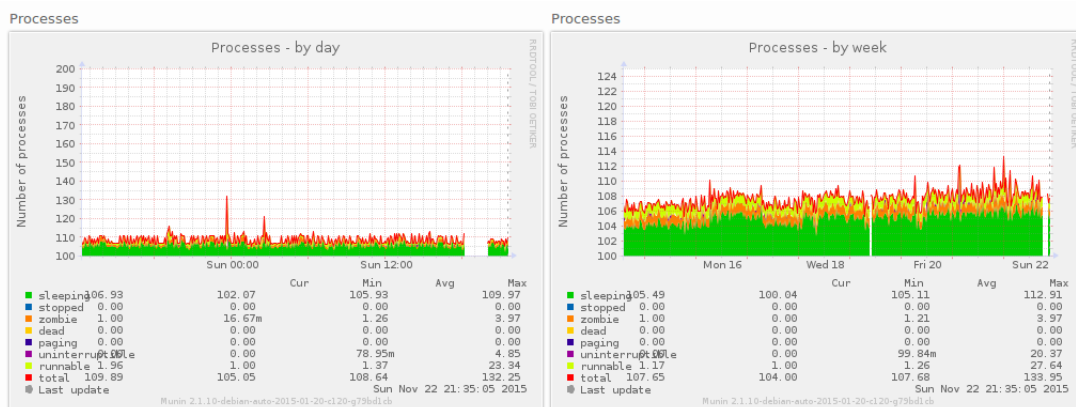


Figura 8.5: Procesos

En estas gráficas se puede ver el numero de procesos del sistema con sus estados, por ejemplo el numero de procesos durmiendo, parados, zombie, corriendo, ininterrumpidos, etc.

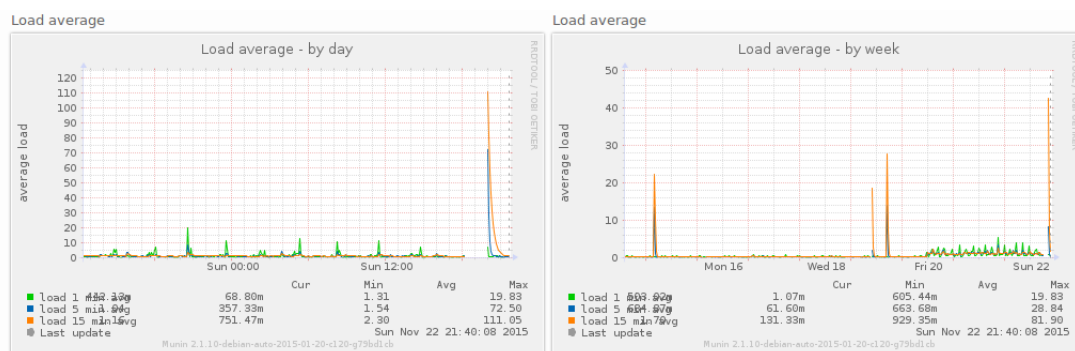


Figura 8.6: Carga media del sistema

En estas gráficas nos muestra la carga media del sistema, de un minuto, 5 minutos y 15 minutos. En definitiva nos muestra la carga media del sistema de forma parecida a como lo hace la orden uptime de linux.

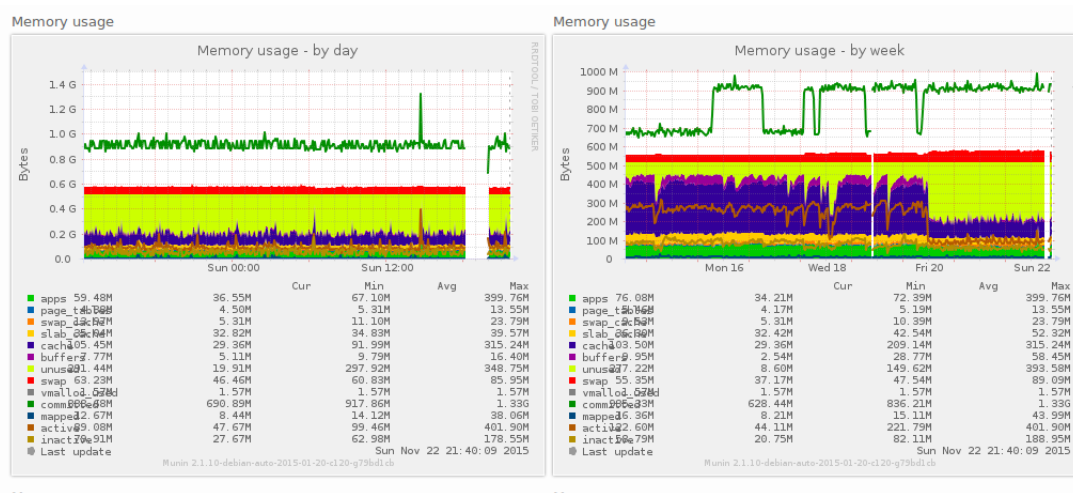


Figura 8.7: Uso de la memoria

Aquí vemos las gráficas de uso de la memoria expresadas en GB, podemos apreciar que durante el fin de semana la memoria cache apenas se ha utilizado en comparación con los días laborables de la semana y que ha habido mucha mas memoria sin usar el fin de semana que durante los días laborables.<sup>5</sup>

<sup>5</sup><http://demo.munin-monitoring.org/munin-monitoring.org/demo.munin-monitoring.org/index.html>

## 9. Haga lo mismo que con Munin

En la remo de ganglia puedes elegir varias opciones a la hora de medir las estadísticas, yo he elegido Analytics cluster eqiad y el nodo analytics1001.eqiad.wmnet

Las primeras gráficas vemos que muestra una panorámica del host mostrando la carga, los bytes de bajada y de subida de la red, la memoria, y el tanto por ciento de uso de la cpu.

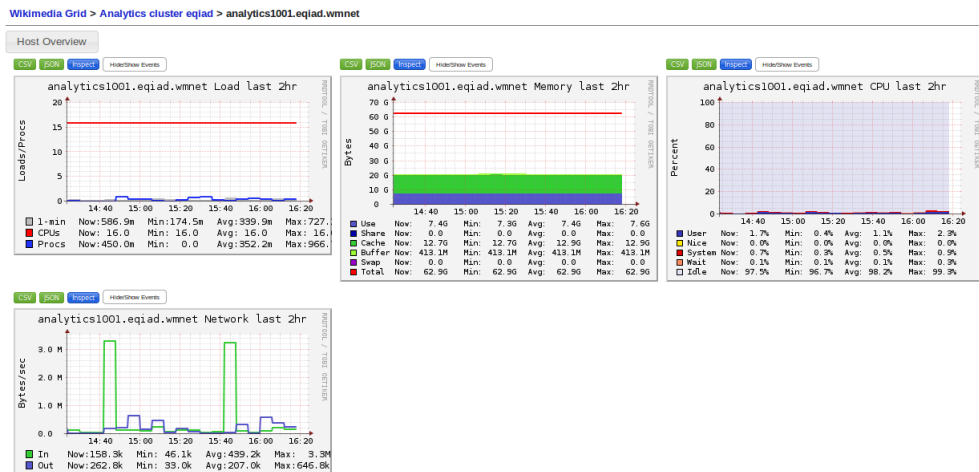


Figura 9.1: Panorámica del host

Mas abajo ofrece la posibilidad de visualizar estadísticas del nodo, por ejemplo las medidas de cpu.



Figura 9.2: Medidas de cpu

Se puede ver el tanto por ciento de uso de la cpu desglosado en las distintas categorías, ociosa, usuario, etc. También se puede ver las estadísticas de disco. Desglosada en espacio de disco disponible, total de espacio de disco y el tanto por ciento de uso de disco.



Figura 9.3: Estadísticas de disco

También se ve las gráficas de la carga media del sistema, tanto de 1 minuto, de 5 minutos como de 15 minutos.



Figura 9.4: Carga media del sistema

También se puede visualizar estadísticas de red y de procesos, de las estadísticas de red podemos ver los bytes enviados y recibidos y los paquetes enviados y recibidos. En cuanto a los procesos se puede ver los procesos totales del sistema y los procesos que están corriendo.



Figura 9.5: Estadísticas de red y de procesos

Por ultimo también nos muestran las medidas de memoria.



Figura 9.6: Estadísticas de la memoria

Como se puede ver vienen desglosadas en memoria de buffers, memoria libre, memoria cache, memoria compartida y memoria de swap.<sup>6</sup>

## 10. Instale el monitor y muestre y comente algunas capturas de pantalla

Para instalarlo ponemos: `yum -enablerepo=epel -y install awstats`, reiniciamos el servicio httpd y generamos el archivo de reportes con `/usr/share/awstats/wwwroot/cgi-bin/awstats.pl -config=localhost.localdomain -create` luego hacemos lo mismo pero con update en vez de create. ahora accedemos a traves del navegador escribiendo `localhost/awstats/awstats.pl?config=localhost.localdomain` Nos sale esta pantalla

<sup>6</sup>[http://ganglia.wikimedia.org/latest/?r=2hr&cs=&ce=&c=Analytics+cluster+eqiad&h=analytics1001.eqiad.wmnet&tab=m&vn=&hide-hf=false&m=network\\_report&sh=1&z=small&hc=0&host\\_regex=&max\\_graphs=10&s=by+name](http://ganglia.wikimedia.org/latest/?r=2hr&cs=&ce=&c=Analytics+cluster+eqiad&h=analytics1001.eqiad.wmnet&tab=m&vn=&hide-hf=false&m=network_report&sh=1&z=small&hc=0&host_regex=&max_graphs=10&s=by+name)

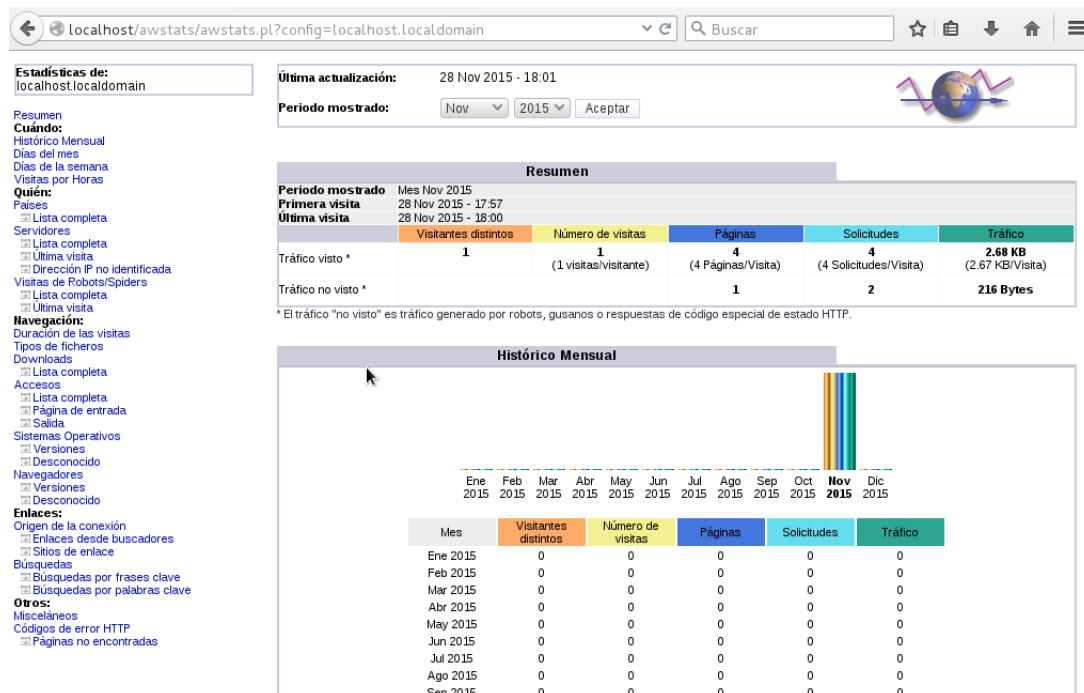


Figura 10.1: Pagina inicial de Awstats

Como se puede ver nos aparece un resumen y histórico mensual, como lo acabo de instalar pues no aparece muchos datos y hay poco que comentar.



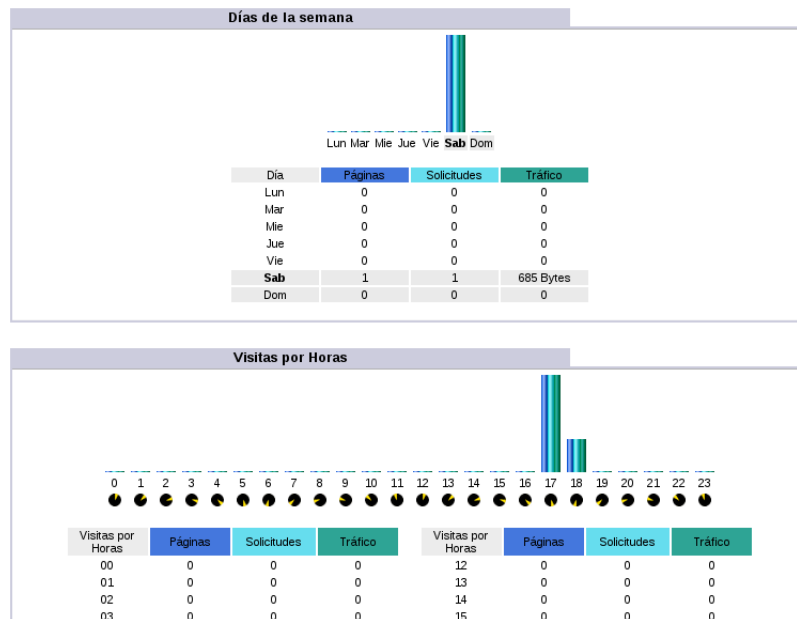


Figura 10.2: Histórico de días de la semana y de horas

En esta imagen nos aparece un histórico de los días de la semana y de las horas.

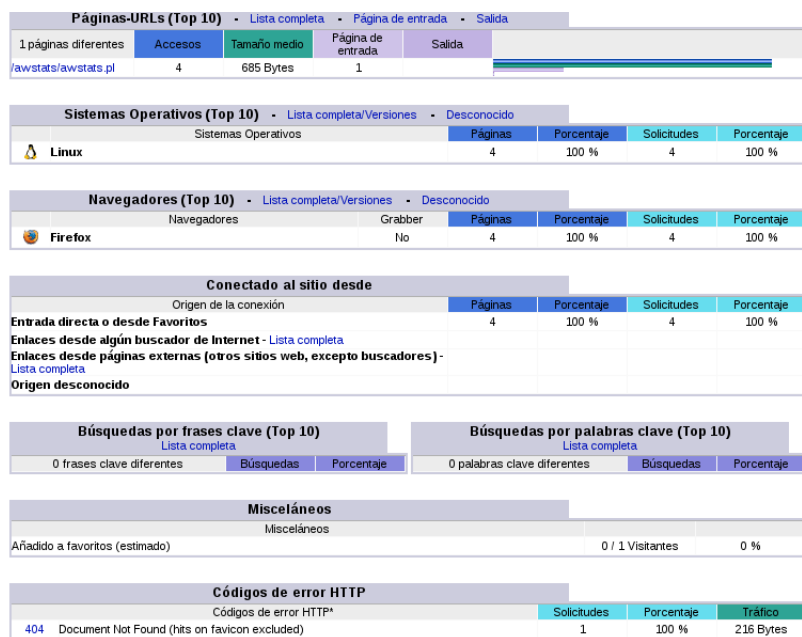


Figura 10.3: Otros datos de awstat

Aquí es muestran algunos datos interesantes. Como las paginas y urls mas consultadas,

sistemas operativos y navegadores usados, búsqueda por frases y por palabras y paginas no encontradas.

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

En el artículo <sup>7</sup> el autor nos explica una anécdota de su carrera profesional en la que dice que strace es una herramienta fundamental y sin duda alguna lo es, por la cantidad de cosas que nos llega a mostrar, básicamente strace es una orden de la línea de comandos que nos muestra los procesos ejecutados por un programa y sus hijos. Esto nos permite desmenuzar el programa y analizar lo que hace al detalle. En el artículo el autor nos muestra un ejemplo utilizando el siguiente código:

```
#include <stdio.h>
void main() {
    printf("hi\n");
}
```

Ahora ejecuta el programa con strace para ello escribe en la línea de comandos *strace -s 2000 -f ./programa*, el -s indica a strace el tamaño de la cadena máxima que tiene que pintar. y el -f muestra los procesos hijos que crea el programa a ejecutar. El autor muestra la salida de strace y se puede ver muchas llamadas al sistema como el *execve()* que indica la ejecución del programa. La salida concluye con un *write()* que imprime el hola y un *exit()* para finalizar el programa. Como se puede ver esta orden es muy potente ya que muestra todo lo que pasa dentro del SO cuando se ejecuta un programa. Personalmente esta orden yo la conocía, la use con un pequeño programa en C que servía como calculadora, el cual tenía que introducir una contraseña, el problema estaba que no me acordaba de la contraseña, además, el código fuente no lo encontraba y en el tenía escrita la contraseña para compararla con la que introducía el usuario. Busque por internet y encontré la orden strace y al ejecutar mi programa se veía mi contraseña verdadera siendo comparada con la que yo introducía. Volviendo al artículo, el autor concluye que strace es una herramienta muy potente y anima al lector a investigar más sobre esta orden, sin duda alguna, y en mi opinión, estoy totalmente de acuerdo con el autor ya que esta herramienta ofrece un muchas opciones y te hace "la vida un poco más fácil, yo creo que todo programador debería de conocer esta orden por su gran utilidad.

## 12. Desarrolle una pagina en C o en C++ y analice su comportamiento usando valdgrind.

```
#include <stdio.h>
```

---

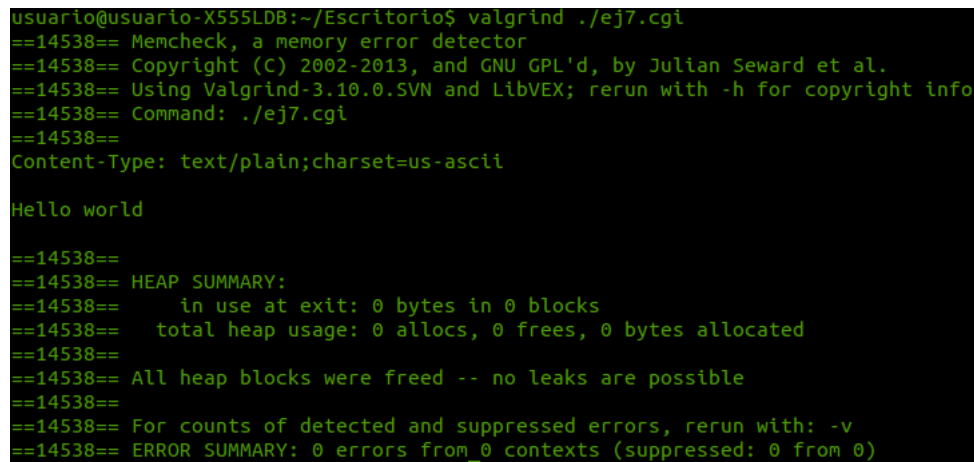
<sup>7</sup><http://chadfowler.com/blog/2014/01/26/the-magic-of-strace/>

```

int main(void) {
    printf("Content-Type: text/plain;charset=us-ascii\n\n");
    printf("Hello world\n\n");
    return 0;
}

```

valgrind ofrece muchos tipos de herramientas, entre las que ofrece esta memcheck: detecta errores en memoria dinámica, cachegrind: permite mejorar la rapidez de ejecución del código, callgrind: da información sobre las llamadas a métodos producidas por el código en ejecución, massif: ayuda a reducir la cantidad de memoria usada por el programa.<sup>8</sup> Si se pone la orden valgrind sin ninguna herramienta específica por defecto utilizará la de memcheck. Ese es un ejemplo de su salida en nuestro programa.



```

usuario@usuario-X555LDB:~/Escritorio$ valgrind ./ej7.cgi
==14538== Memcheck, a memory error detector
==14538== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al.
==14538== Using Valgrind-3.10.0.SVN and LibVEX; rerun with -h for copyright info
==14538== Command: ./ej7.cgi
==14538==
Content-Type: text/plain;charset=us-ascii

Hello world

==14538==
==14538== HEAP SUMMARY:
==14538==   in use at exit: 0 bytes in 0 blocks
==14538==   total heap usage: 0 allocs, 0 frees, 0 bytes allocated
==14538==
==14538== All heap blocks were freed -- no leaks are possible
==14538==
==14538== For counts of detected and suppressed errors, rerun with: -v
==14538== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)

```

Figura 12.1: Ejecución de valgrind

### 13. Escriba un script en python y analice su comportamiento usando el profiler presentado

```

lista=list(range(100000000))

for i in lista:
    i*=2147483647
for i in lista:
    i%=13395

lista.sort(reverse=True)

```

Ejecutamos el script con el profiler.

<sup>8</sup><http://valgrind.org/info/tools.html>

```

usuario@usuario-X555LDB:~/Informatica/3/ISE/Practicas/P3$ python -m cProfile sc
ipt.py
      4 function calls in 17.120 seconds

Ordered by: standard name

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
      1    14.564    14.564    17.120    17.120 script.py:4(<module>)
      1     0.000     0.000     0.000     0.000 {method 'disable' of '_lsprof.Pro
filer' objects}
      1     1.398     1.398     1.398     1.398 {method 'sort' of 'list' objects}
      1     1.159     1.159     1.159     1.159 {range}

```

Figura 13.1: Ejecucion del profiler

CProfile se puede llamar dentro del codigo y hacer el profiling de una funcion o de un trozo de un codigo tan solo con un import. Pero yo la he llamado para ejecutar el script entero. Los datos que nos ofrecen son el ncalls: para el numero de llamadas. tottime: para el total de tiempo gastado en la funcion, esto solo es para el uso en subrutinas. percall: cociente de tottime dividido entre ncalls. cumtime: tiempo acumulativo gastando en la funcion y sus subrutinas. percall: es el cociente de cumtime dividido por las llamadas sus llamadas. filename:lineno(function): provee los datos respectivos a cada funcion.<sup>9</sup>

## 14. Acceda a la consola de mysql y muestre el resultado de mostrar el profile de una consulta

Primero creamos la tabla e insertamos los datos:

```
CREATE TABLE persona(nombre VARCHAR(40), apellido VARCHAR(40), edad NUMERIC );
```

Figura 14.1: Creacion de la tabla

```
insert into persona values ('Antonio','Cabrera',34);
```

Figura 14.2: Insercion de datos 1

```
insert into persona values ('Manuel','Sanchez',35);
```

Figura 14.3: Insercion de datos 2

```
insert into persona values ('Pedro','Mayas',19);
```

Figura 14.4: Insercion de datos 3

<sup>9</sup><https://docs.python.org/2/library/profile.html>

Para activar el profiler, debemos de poner en la linea de comandos *mysql set profiling=1;*  
Ahora todas las consultas que hagamos seran medidas.

```
mysql> set profiling=1;
Query OK, 0 rows affected, 1 warning (0,00 sec)

mysql> select nombre, apellido from persona where edad=34;
+-----+-----+
| nombre | apellido |
+-----+-----+
| Antonio | Cabrera |
+-----+-----+
1 row in set (0,00 sec)
```

Figura 14.5: Activacion del profilin y consulta sql

Ahora podemos ver el tiempo de cada consulta que hagamos con *show profiles*.

```
mysql> show profiles;
+-----+-----+-----+
| Query_ID | Duration | Query |
+-----+-----+-----+
| 1 | 0.00096225 | select nombre, apellido from persona where edad=34 |
+-----+-----+-----+
```

Figura 14.6: Tiempo de la consulta

si queremos ver una en concreto ponemos *show profile for query* y el numero de la consulta, como nosotros hemos hecho solo una ponemos el numero 1.<sup>10</sup>

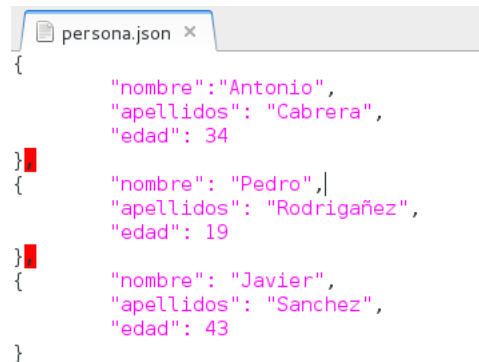
```
mysql> show profile for query 1;
+-----+-----+
| Status | Duration |
+-----+-----+
| starting | 0.000163 |
| checking permissions | 0.000019 |
| Opening tables | 0.000035 |
| init | 0.000062 |
| System lock | 0.000016 |
| optimizing | 0.000025 |
| statistics | 0.000042 |
| preparing | 0.000028 |
| executing | 0.000005 |
| Sending data | 0.000128 |
| end | 0.000011 |
| query end | 0.000013 |
| closing tables | 0.000017 |
| freeing items | 0.000027 |
| cleaning up | 0.000033 |
+-----+-----+
15 rows in set, 1 warning (0,03 sec)
```

Figura 14.7: Visualizacion de los tiempos medidos de la consulta

<sup>10</sup><http://rooteando.com/mysql-profiler-conoce-el-rendimiento-de-tus-consultas/>

## 15. Al igual que ha realizado el profiling con MySQL, realice lo mismo con MongoDB y compare los resultados (use la misma informacion y la misma consulta)

Primero creamos las tablas iguales que en sql, con tres filas y tres campos.



```
{
  "nombre": "Antonio",
  "apellidos": "Cabrera",
  "edad": 34
},
{
  "nombre": "Pedro",
  "apellidos": "Rodriguez",
  "edad": 19
},
{
  "nombre": "Javier",
  "apellidos": "Sanchez",
  "edad": 43
}
```

Figura 15.1: Creacion de tablas

Luego importamos el archivo json.

```
[ajavier@localhost Descargas]$ mongoimport --host localhost --port 27017 --db personas --collection gente --file
/home/ajavier/Descargas/persona.json --jsonArray
```

Figura 15.2: Importacion del archivo json

Debemos de activar el profiler. Por defecto mongodb hace un profiling de las consultas que tardan mucho, pero podemos modificarlo poniendo el modo profiling nivel 2 para que haga profiling de todas las consultas. Para poner nivel dos ponemos `db.setProfilingLevel(2)` Ahora hacemos la consulta igual que la del ejercicio anterior pero ahora con mongo.<sup>11</sup>

```
> db.gente.find({"edad": 34 }, {"nombre": 1, "apellidos": 1, "_id": 0});
{ "nombre" : "Antonio", "apellidos" : "Cabrera" }
```

Figura 15.3: Consulta

Una vez hecha la consulta el resultado del profiling lo guarda en la coleccion system.profile, debemos de hacer una consulta en esta coleccion para que nos diga el resultado de nuestra anterior consulta. Cambiamos a la base de datos system.profile y hacemos la consulta buscando nuestra consulta anterior.

<sup>11</sup><https://docs.mongodb.org/manual/tutorial/manage-the-database-profiler/>

```
> db.system.profile.find({"op":"query", "ns" : "personas.gente"})
{ "op" : "query", "ns" : "personas.gente", "query" : { "edad" : 34 }, "ntoreturn" : 0, "ntoskip" : 0, "nscanned" : 3, "nscannedObjects" : 3, "keyUpdates" : 0, "numYield" : 0, "lockStats" : { "timeLockedMicros" : { "r" : NumberLong(87188), "w" : NumberLong(0) }, "timeAcquiringMicros" : { "r" : NumberLong(5), "w" : NumberLong(9) } }, "nreturned" : 1, "responseLength" : 68, "millis" : 87, "execStats" : { "type" : "PROJECTION", "works" : 5, "yields" : 1, "unyields" : 1, "invalidates" : 0, "advanced" : 1, "needTime" : 0, "needFetch" : 0, "isEOF" : 1, "children" : [ { "type" : "COLLSCAN", "works" : 5, "yields" : 1, "unyields" : 1, "invalidates" : 0, "advanced" : 1, "needTime" : 3, "needFetch" : 0, "isEOF" : 1, "docsTested" : 3, "children" : [ ] } ] }, "ts" : ISODate("2015-11-22T23:30:40.196Z"), "client" : "127.0.0.1", "allUsers" : [ ], "user" : "" }
```

Figura 15.4: Profiling MongoDB

Buscamos por operacion query, ns personas.gente para que nos salga la nuestra.