

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

---

## Memoria Práctica

---

Ángel Valera Motos

30 de noviembre de 2014

## Índice

|  |    |
|--|----|
| 1. ¿Qué archivo le permite ver qué programas se han instalado con el gestor de paquetes?   | 4  |
| 1.1. ¿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)  | 5  |
| 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.   | 6  |
| 4. Ejecute el monitor de "System Performance" y muestre el resultado. Incluya capturas de pantalla comentando la información que aparece.  | 7  |
| 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). alternatively, 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 ( <a href="http://demo.munin-monitoring.org/">http://demo.munin-monitoring.org/</a> ) donde se muestra cómo monitorizan un servidor. Monitorice varios parámetros y haga capturas de pantalla de lo que está mostrando comentando qué observa.   | 13 |
| 8. Escribe un breve resumen sobre alguno de los artículos donde se muestra el uso de strace o busque otro y comentelo  | 21 |
| 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)  | 22 |

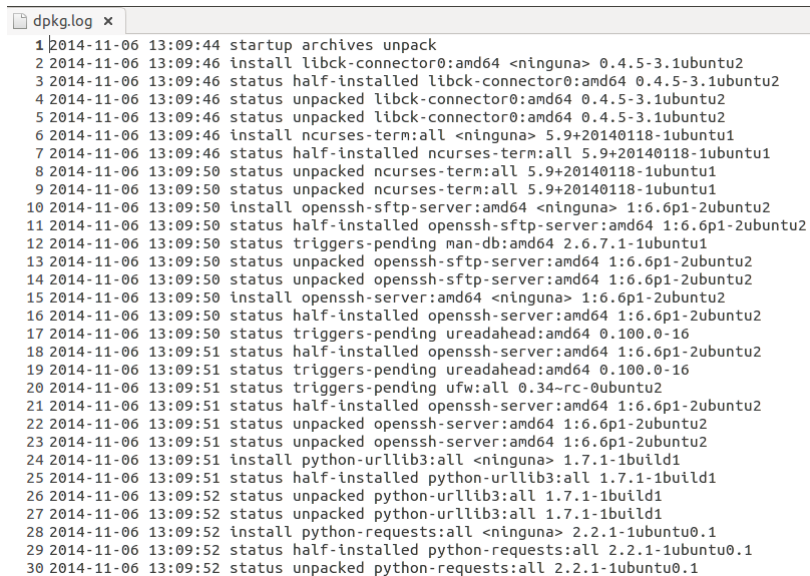
## Índice de figuras

|  |    |
|--|----|
| 1.1. Contenido del archivo /var/log/dpkg.log . . . . .                             | 4  |
| 3.1. Ejecución de demesg antes de conectar el dispositivo USB . . . . .            | 6  |
| 3.2. Ejecución de demesg antes de conectar el dispositivo USB . . . . .            | 6  |
| 4.1. Ejecución del monitor System Performance . . . . .                            | 7  |
| 5.1. Creación de un nuevo recopilador de datos . . . . .                           | 8  |
| 5.2. Selección de los datos del recopilador . . . . .                              | 9  |
| 5.3. Agregamos los contadores del recopilador . . . . .                            | 9  |
| 5.4. Indicamos el tiempo entre cada muestra del recopilador . . . . .              | 10 |
| 5.5. Selección de la carpeta donde guardar las muestras del recopilador . . . . .  | 10 |
| 5.6. Ejemplo de ejecución de nuestro nuevo recopilador . . . . .                   | 11 |
| 6.1. Ejemplo de ejecución de CPUID . . . . .                                       | 12 |
| 6.2. Ejemplo de ejecución de CoreTemp . . . . .                                    | 12 |
| 7.1. Ejemplo monitorización del sistema con Munin . . . . .                        | 13 |
| 7.2. Ejemplo monitorización de apache con Munin . . . . .                          | 14 |
| 7.3. Ejemplo monitorización de los procesos del servidor con Munin . . . . .       | 15 |
| 7.4. Ejemplo de monitorización de Ganglia . . . . .                                | 16 |
| 7.5. Instalación de cacti I . . . . .  | 17 |
| 7.6. Instalación de cacti II . . . . .   | 17 |
| 7.7. Instalación de cacti III . . . . .  | 18 |
| 7.8. Interfaz de acceso a cacti . . . . .  | 18 |
| 7.9. Interfaz de cacti . . . . .   | 19 |
| 7.10. Ejemplo de ejecución de cacti I . . . . .                                    | 19 |
| 7.11. Monitorización de la memoria usada tanto libre como de intercambio . . . . . | 20 |
| 7.12. Muestras con la media de carga y los usuarios que han logueado . . . . .     | 20 |
| 9.1. Perfil de una consulta sencilla en phpmyadmin I . . . . .                     | 22 |
| 9.2. Perfil de una consulta sencilla en phpmyadmin II . . . . .                    | 23 |

## Índice de tablas

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

El archivo que nos permite ver qué programas se han instalado con el gestor de paquetes lo encontramos en `/var/log/dpkg.log` el cual registra que paquetes se instalaron y/o desinstalaron en el sistema. También podemos ver los programas instalados con el gestor de paquetes en el archivo `/var/log/apt/history.log` ya que es apt el gestor de paquetes (que a su vez usa dpkg). En la figura 1.1 podemos ver el contenido de `dpkg.log`.



```
dpkg.log x
1 2014-11-06 13:09:44 startup archives unpack
2 2014-11-06 13:09:46 install libck-connector0:amd64 <ninguna> 0.4.5-3.1ubuntu2
3 2014-11-06 13:09:46 status half-installed libck-connector0:amd64 0.4.5-3.1ubuntu2
4 2014-11-06 13:09:46 status unpacked libck-connector0:amd64 0.4.5-3.1ubuntu2
5 2014-11-06 13:09:46 status unpacked libck-connector0:amd64 0.4.5-3.1ubuntu2
6 2014-11-06 13:09:46 install ncurses-term:all <ninguna> 5.9+20140118-1ubuntu1
7 2014-11-06 13:09:46 status half-installed ncurses-term:all 5.9+20140118-1ubuntu1
8 2014-11-06 13:09:50 status unpacked ncurses-term:all 5.9+20140118-1ubuntu1
9 2014-11-06 13:09:50 status unpacked ncurses-term:all 5.9+20140118-1ubuntu1
10 2014-11-06 13:09:50 install openssh-sftp-server:amd64 <ninguna> 1:6.6p1-2ubuntu2
11 2014-11-06 13:09:50 status half-installed openssh-sftp-server:amd64 1:6.6p1-2ubuntu2
12 2014-11-06 13:09:50 status triggers-pending man-db:amd64 2.6.7.1-1ubuntu1
13 2014-11-06 13:09:50 status unpacked openssh-sftp-server:amd64 1:6.6p1-2ubuntu2
14 2014-11-06 13:09:50 status unpacked openssh-sftp-server:amd64 1:6.6p1-2ubuntu2
15 2014-11-06 13:09:50 install openssh-server:amd64 <ninguna> 1:6.6p1-2ubuntu2
16 2014-11-06 13:09:50 status half-installed openssh-server:amd64 1:6.6p1-2ubuntu2
17 2014-11-06 13:09:50 status triggers-pending ureadahead:amd64 0.100.0-16
18 2014-11-06 13:09:51 status half-installed openssh-server:amd64 1:6.6p1-2ubuntu2
19 2014-11-06 13:09:51 status triggers-pending ureadahead:amd64 0.100.0-16
20 2014-11-06 13:09:51 status triggers-pending ufw:all 0.34-rc-0ubuntu2
21 2014-11-06 13:09:51 status half-installed openssh-server:amd64 1:6.6p1-2ubuntu2
22 2014-11-06 13:09:51 status unpacked openssh-server:amd64 1:6.6p1-2ubuntu2
23 2014-11-06 13:09:51 status unpacked openssh-server:amd64 1:6.6p1-2ubuntu2
24 2014-11-06 13:09:51 install python-urllib3:all <ninguna> 1.7.1-1build1
25 2014-11-06 13:09:51 status half-installed python-urllib3:all 1.7.1-1build1
26 2014-11-06 13:09:52 status unpacked python-urllib3:all 1.7.1-1build1
27 2014-11-06 13:09:52 status unpacked python-urllib3:all 1.7.1-1build1
28 2014-11-06 13:09:52 install python-requests:all <ninguna> 2.2.1-1ubuntu0.1
29 2014-11-06 13:09:52 status half-installed python-requests:all 2.2.1-1ubuntu0.1
30 2014-11-06 13:09:52 status unpacked python-requests:all 2.2.1-1ubuntu0.1
```

Figura 1.1: Contenido del archivo `/var/log/dpkg.log`

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

Las terminaciones `.1.gz` y `.2.gz` [6] sirven para comprimir archivos como el `dpkg.log` pues con esos archivos llega un momento en el que se llenan, entonces lo que se hace es copiarlo en un archivo `dpkg.log.1.gz` que lo comprime y se borra el contenido del `dpkg.log` que puede volver a registrar que paquetes se instalaron y/o desinstalaron en el sistema. De la misma forma cuando se vuelva a llenar se realizará el mismo proceso pero ahora se llamará `dpkg.log.2.gz`, ..., `3.gz`, etc.

En mi configuración la primera vez que se llena (`.1`) no comprime sino que comienza a comprimir a partir de la segunda vez (`.2.gz`).

## 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)

Para programar una tarea [1] que se desea que se ejecute se hace construyendo un archivo (llamado archivo crontab") que deberá tener un determinado formato, (llamado formato crontab"). Será con la orden crontab con la que indicamos el archivo con formato crontab" que deseamos comunicar al demonio cron.

Cada línea de código de un archivo crontab (excepto los comentarios que están precedidos por #) pueden contener estos campos (que representan una orden):

**minuto hora día-del-mes mes día-de-la-semana orden**

Para realizar el ejemplo primero creamos un script (llamado ej2) con el siguiente contenido:

```
#!/bin/bash
```

```
cp -r /home/angelvalera/codigo /home/angelvalera/seguridad-`date +%d-%m-%y`
```

Ahora nos creamos un archivo crontab (para ello nos podemos fijar en el apartado EXAMPLE CRON FILE del manual de crontab [8]) llamado (crontab 1) con el siguiente contenido:

```
00 12 * * * ./ej2
```

De esta manera realizamos una copia de seguridad del directorio código todos los días a las 12:00. Para ejecutarlo hacemos:

**crontab crontab1**

y para ver los crontab que tenemos :

**crontab -l**

### 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.

Usando el comando dmesg [9] antes de conectar un dispositivo USB, se nos muestra lo último acontecido en el sistema como podemos ver en la figura 3.1

```
angelvalera@angelvalera-VirtualBox:~$ dmesg | tail
[ 39.585426] ip_tables: (c) 2000-2006 Netfilter Core Team
[ 44.371232] init: plymouth-stop pre-start process (1676) terminated with status 1
[ 47.474150] init: anacron main process (833) killed by TERM signal
[ 53.861859] audit_printk_skb: 174 callbacks suppressed
[ 53.861866] type=1400 audit(1417012993.297:70): apparmor="STATUS" operation="profile_replace" profile="unconfined" name="/usr/lib/cups/backen
d/cups-pdf" pid=2026 comm="apparmor_parser"
[ 53.861877] type=1400 audit(1417012993.297:71): apparmor="STATUS" operation="profile_replace" profile="unconfined" name="/usr/sbin/cupsd" pid
=2026 comm="apparmor_parser"
[ 53.862590] type=1400 audit(1417012993.297:72): apparmor="STATUS" operation="profile_replace" profile="unconfined" name="/usr/sbin/cupsd" pid
=2026 comm="apparmor_parser"
[ 168.535746] ISO 9660 Extensions: Microsoft Joliet Level 3
[ 168.553234] ISO 9660 Extensions: RRIP_1991A
[ 189.024422] systemd-hostnamed[2722]: Warning: nss-myhostname is not installed. Changing the local hostname might make it unresolvable. Pleas
e install nss-myhostname!
```

Figura 3.1: Ejecución de demesg antes de conectar el dispositivo USB

Después de conectar el dispositivo podemos ver como al ejecutar demegs nos muestra información acerca de ese dispositivo, como podemos ver en la figura 3.2

```
[ 1192.712433] usb 1-2: new full-speed USB device number 4 using ohci-pci
[ 1193.378051] usb 1-2: New USB device found, idVendor=058f, idProduct=6387
[ 1193.378062] usb 1-2: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 1193.378066] usb 1-2: Product: Mass Storage Device
[ 1193.378070] usb 1-2: Manufacturer: JetFlash
[ 1193.378075] usb 1-2: SerialNumber: PTD7Y03T
[ 1193.758346] usb-storage 1-2:1.0: USB Mass Storage device detected
[ 1193.766280] scsi3 : usb-storage 1-2:1.0
[ 1193.766673] usbcore: registered new interface driver usb-storage
[ 1194.788193] scsi 3:0:0:0: Direct-Access JetFlash Transcend 8GB 8.07 PQ: 0 ANSI: 2
[ 1194.789990] sd 3:0:0:0: Attached scsi generic sg2 type 0
[ 1194.824368] sd 3:0:0:0: [sdb] 15687680 512-byte logical blocks: (8.03 GB/7.48 GiB)
[ 1194.833939] sd 3:0:0:0: [sdb] Write Protect is off
[ 1194.833960] sd 3:0:0:0: [sdb] Mode Sense: 03 00 00 00
[ 1194.842575] sd 3:0:0:0: [sdb] No Caching mode page found
[ 1194.842590] sd 3:0:0:0: [sdb] Assuming drive cache: write through
[ 1194.899702] sd 3:0:0:0: [sdb] No Caching mode page found
[ 1194.899710] sd 3:0:0:0: [sdb] Assuming drive cache: write through
[ 1195.237891] sdb: sdb1
[ 1195.466113] sd 3:0:0:0: [sdb] No Caching mode page found
[ 1195.466125] sd 3:0:0:0: [sdb] Assuming drive cache: write through
[ 1195.466132] sd 3:0:0:0: [sdb] Attached SCSI removable disk
[ 1198.624908] systemd-hostnamed[3164]: Warning: nss-myhostname is not installed. Changing the local hostname might make it unresolvable. Pleas
e install nss-myhostname!
[ 2776.166807] systemd-hostnamed[3427]: Warning: nss-myhostname is not installed. Changing the local hostname might make it unresolvable. Pleas
e install nss-myhostname!
```

Figura 3.2: Ejecución de demesg antes de conectar el dispositivo USB

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

Para la ejecución del monitor solo tenemos que hacer click derecho y seleccionarla opción iniciar.

Los resultados se guardarán en el directorio que el monitor tenga asignado.

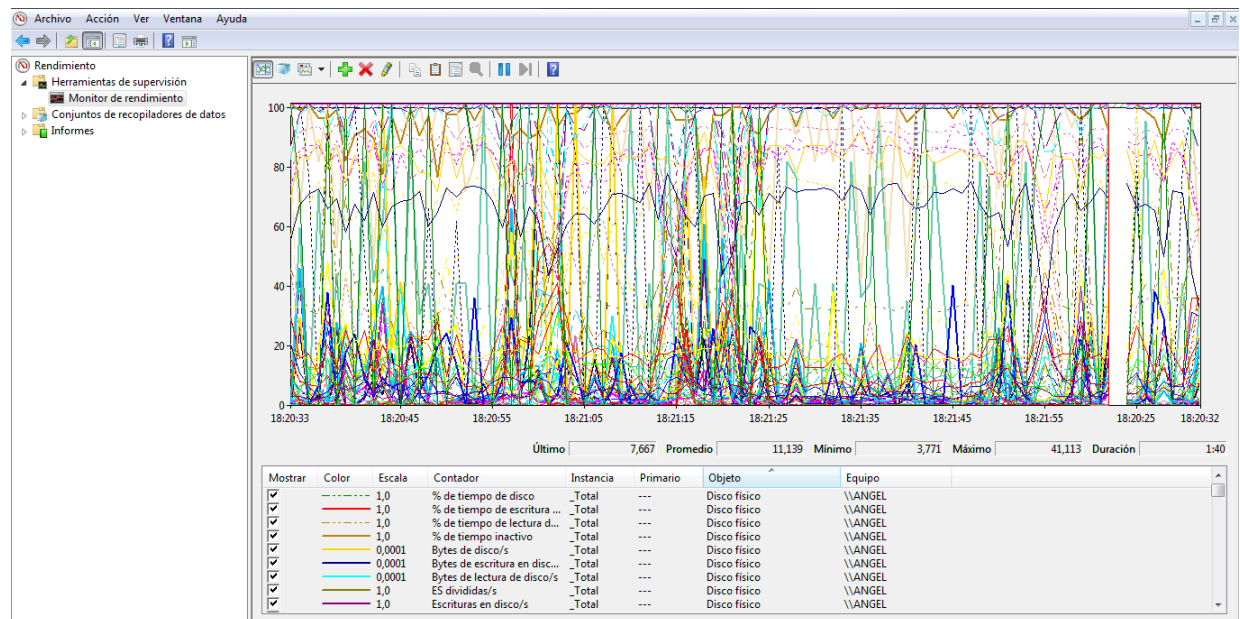


Figura 4.1: Ejecución del monitor System Performance

En la figura 4.1 se nos muestran los parámetros monitorizados de los discos físicos durante un determinado periodo de tiempo. En él podemos ver que monitorizamos el % de tiempo de disco y como va variando durante el intervalo de tiempo, también podemos ver el promedio de % como su máximo y mínimo.

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 nuevo recopilador de datos definido por el usuario, en la pestaña de recopiladores de datos definidos por el usuario le damos a crear nuevo recopilador, indicando primero el nombre y la opción de configuración que en nuestro caso es avanzada, como aparece en la figura 5.1.

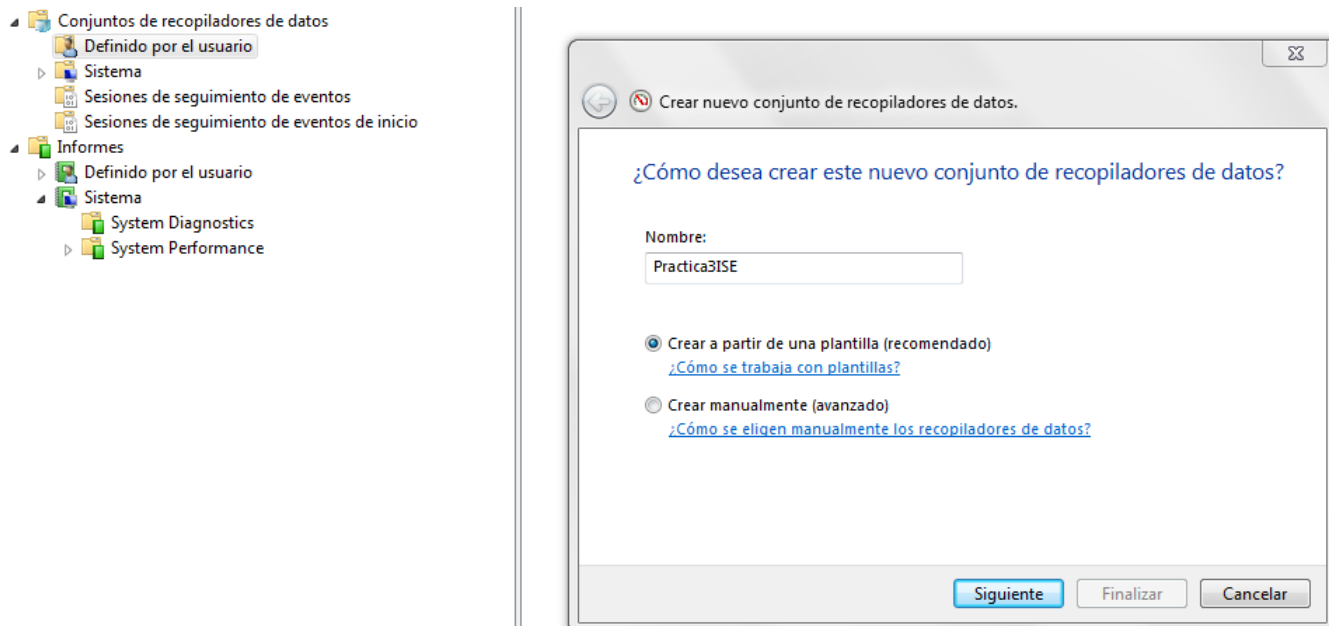


Figura 5.1: Creación de un nuevo recopilador de datos

Como podemos ver en la figura 5.2, seleccionamos los datos a incluir en nuestro monitor.



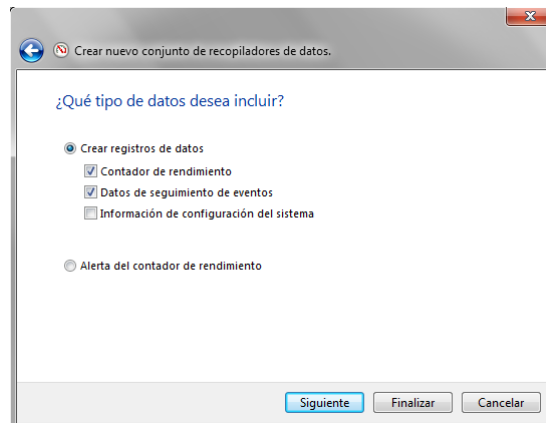


Figura 5.2: Selección de los datos del recopilador

Ahora agregamos los contadores indicados (como no me aparecía servidor web he puesto servidor), como podemos ver en la figura 5.3

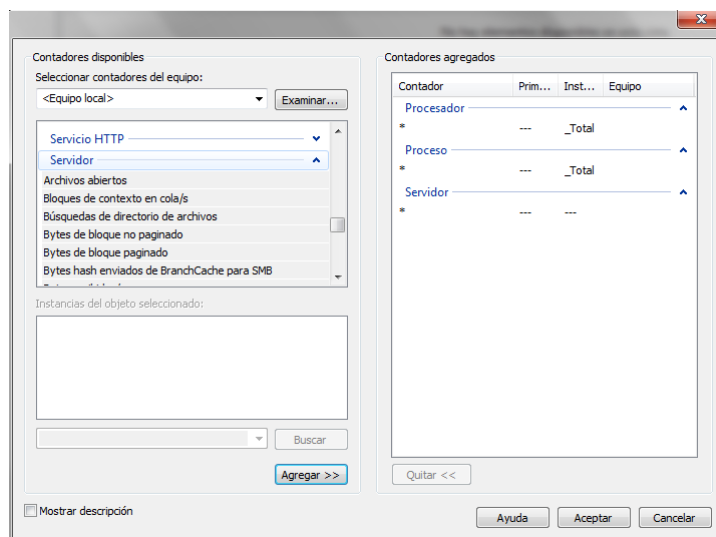


Figura 5.3: Agregamos los contadores del recopilador

Indicamos el tiempo entre cada muestra, en nuestro caso será de 15 segundos, como podemos ver en la figura 5.4.

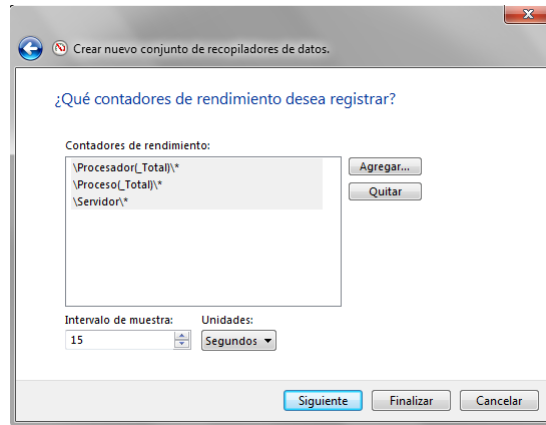


Figura 5.4: Indicamos el tiempo entre cada muestra del recopilador

Por último indicamos la carpeta donde se guardarán los resultados de las muestras que tome nuestro monitor, como aparece en la figura 5.5.

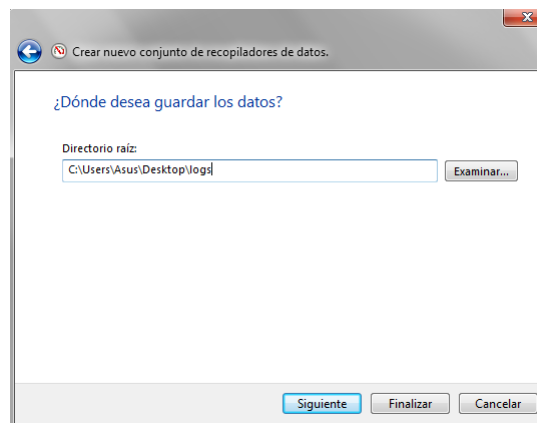


Figura 5.5: Selección de la carpeta donde guardar las muestras del recopilador

Una vez creado nuestro monitor, pasamos a ejecutarlo como vemos en la figura 5.6.

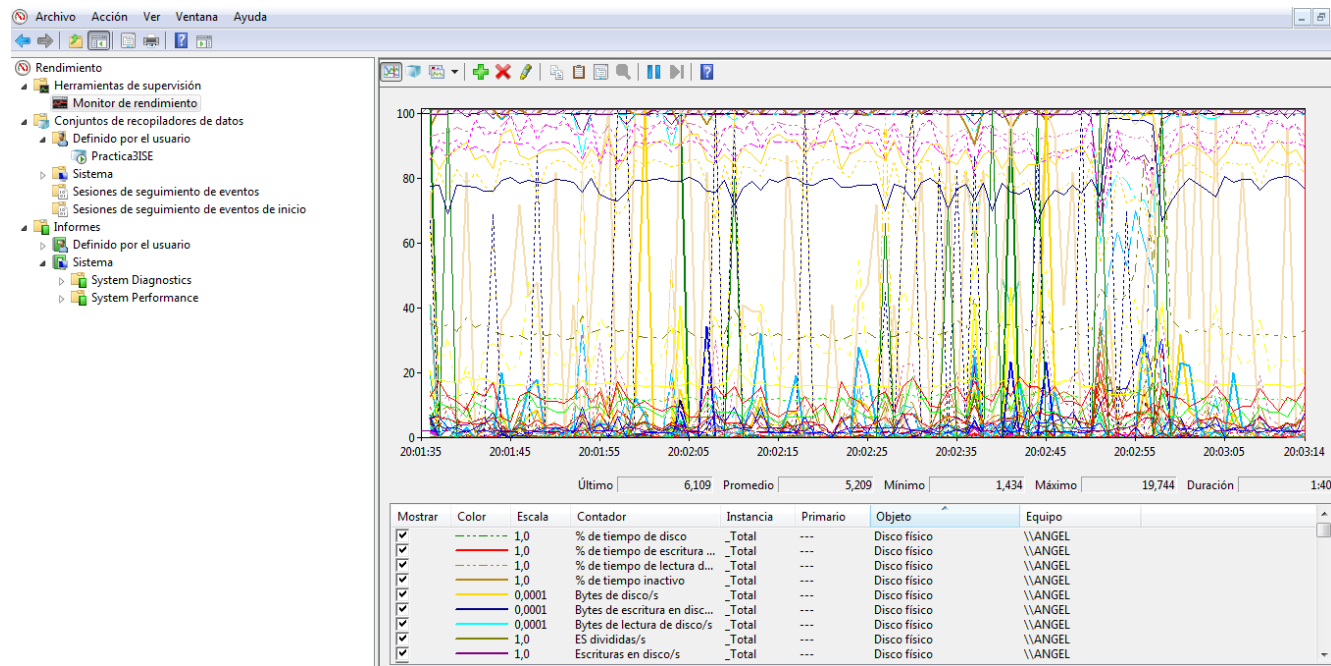


Figura 5.6: Ejemplo de ejecución de nuestro nuevo recopilador

## 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

He instalado dos de los monitores que se comentan, CoreTemp[5] y CPUID[4].

En la figura 6.1 podemos ver un ejemplo de CPUID el cual nos muestra la temperatura y los voltajes de la placa base, procesadores, discos duros y tarjeta gráfica con sus valores mínimos y máximos.

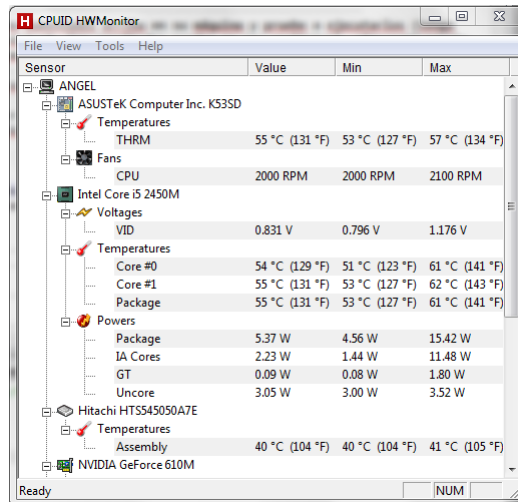


Figura 6.1: Ejemplo de ejecución de CPUID

El otro monitor es CoreTemp, este monitor como vemos en la figura 6.2 ofrece menos información que CPUID, pero en él podemos ver el modelo de procesador que tiene nuestro sistema, la frecuencia a la que van y la temperatura que tiene cada uno de ellos.

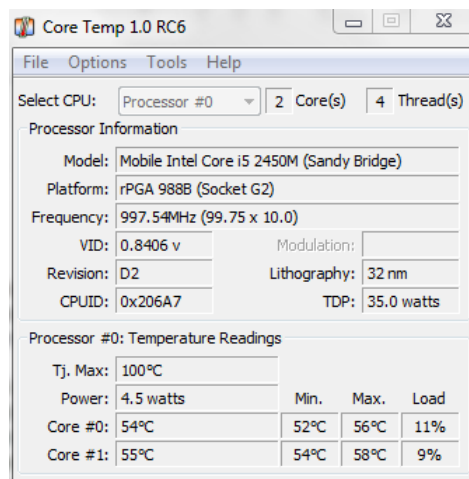


Figura 6.2: Ejemplo de ejecución de CoreTemp

Otros monitores hardware que podemos encontrar son :

**Windows:**

-Aida 64 <http://www.aida64.com/>

-PRTG [http://www.es.paessler.com/monitor\\_hardware](http://www.es.paessler.com/monitor_hardware)

-argus monitor <http://www.argotronic.com/en/index.php>

## Linux:

-Mediante el comando sar podremos realizar monitorización de cpu.

-Monitorix <http://www.monitorix.org/>

## 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. Monitoree varios parámetros y haga capturas de pantalla de lo que está mostrando comentando qué observa.

Como se muestra en la figura 7.1, podemos ver la monitorización desde el punto de vista del sistema, mostrando el porcentaje de uso de los cuatro núcleos de CPU que tiene dicho sistema.

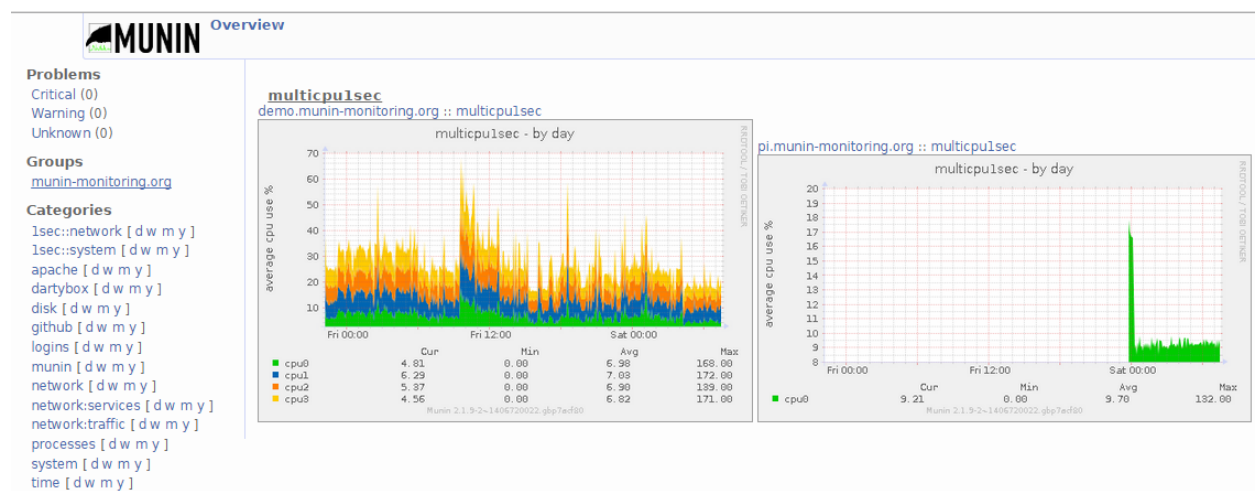


Figura 7.1: Ejemplo monitorización del sistema con Munin

También podemos monitorizar por ejemplo apache, como muestra la figura 7.2 en el que podemos ver los accesos por segundo que hemos tenido en el servidor y los procesos que hemos tenido en el puerto 80.



Figura 7.2: Ejemplo monitorización de apache con Munin

También podemos monitorizar los procesos del servidor y ver el número que hay y en qué estado se encuentran si están dormidos, en estado zombie, ejecutables o muertos por ejemplo, además de sus valores máximos, mínimos y una media, como muestra la figura 7.3.

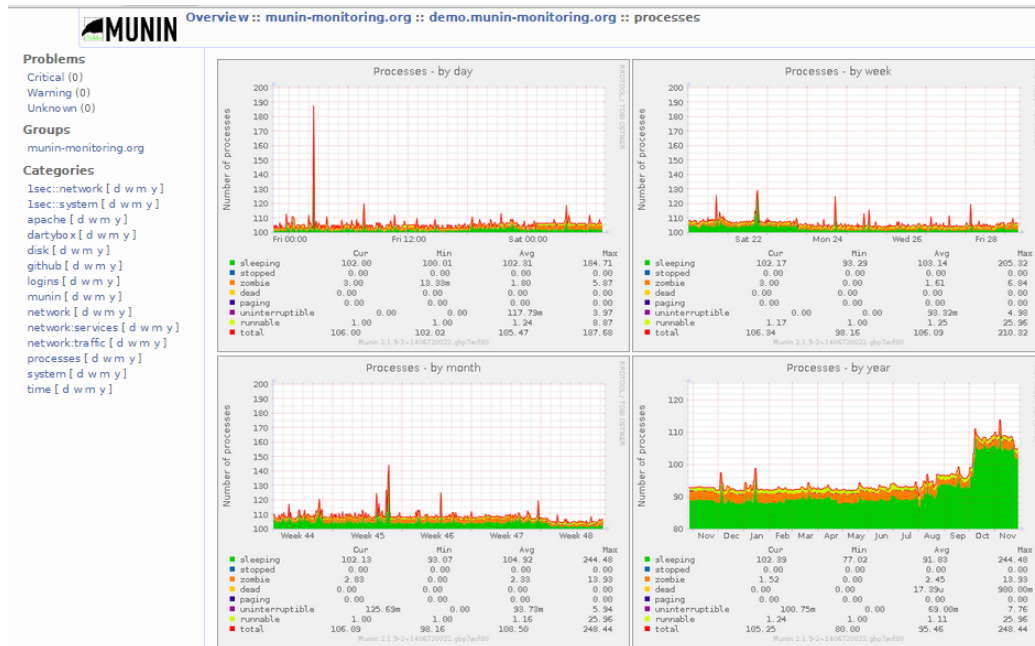


Figura 7.3: Ejemplo monitorización de los procesos del servidor con Munin

### [Cuestión Opcional 3] Haga lo mismo con ganglia que con Munin

Como podemos ver en la figura 7.4 en la primera gráfica se muestra los tiempos de carga de los procesos. En ella se pinta de gris aquellos que han tardado menos de un minuto en procesarse, en rojo se muestran el total de CPUs que hay, también podemos ver el número de host activos y caídos así como la media de tiempo de espera con 3 tiempos de duración (este tiempo de duración es de 15,51 minutos). También muestra otras gráficas que ofrecen información acerca de la memoria.

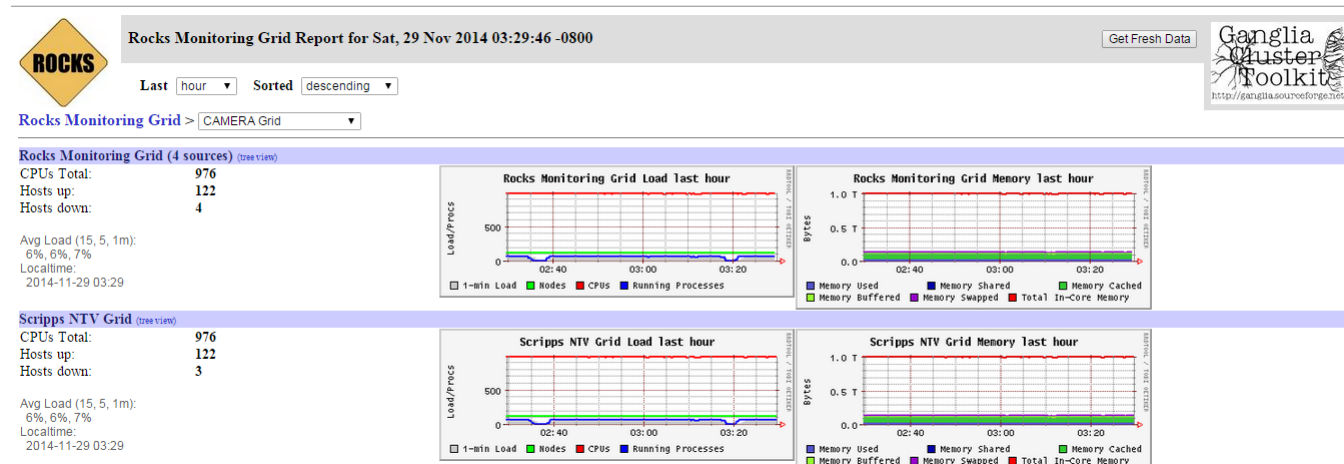


Figura 7.4: Ejemplo de monitorización de Ganglia

## [Cuestión Opcional 5] Pruebe a instalar este monitor en alguno de sus tres sistemas. Realice capturas de pantalla del proceso de instalación y comente las capturas de pantalla del programa de ejecución

Para instalar cacti[7] primero tenemos que tener instalados los paquetes de apache2 php5 mysql-server y phpmyadmin:

escribimos `sudo apt-get install apache2 php5 mysql-server phpmyadmin`

Una vez instalados ahora pasamos a instalar cacti, para eso escribimos:

`sudo apt-get install cacti cacti-spine`

Hecho esto ahora nos vamos al buscador y escribimos `localhost/cacti` para continuar la instalación, si todo fue bien hasta aquí nos debería salir lo que se muestra en la figura 7.5



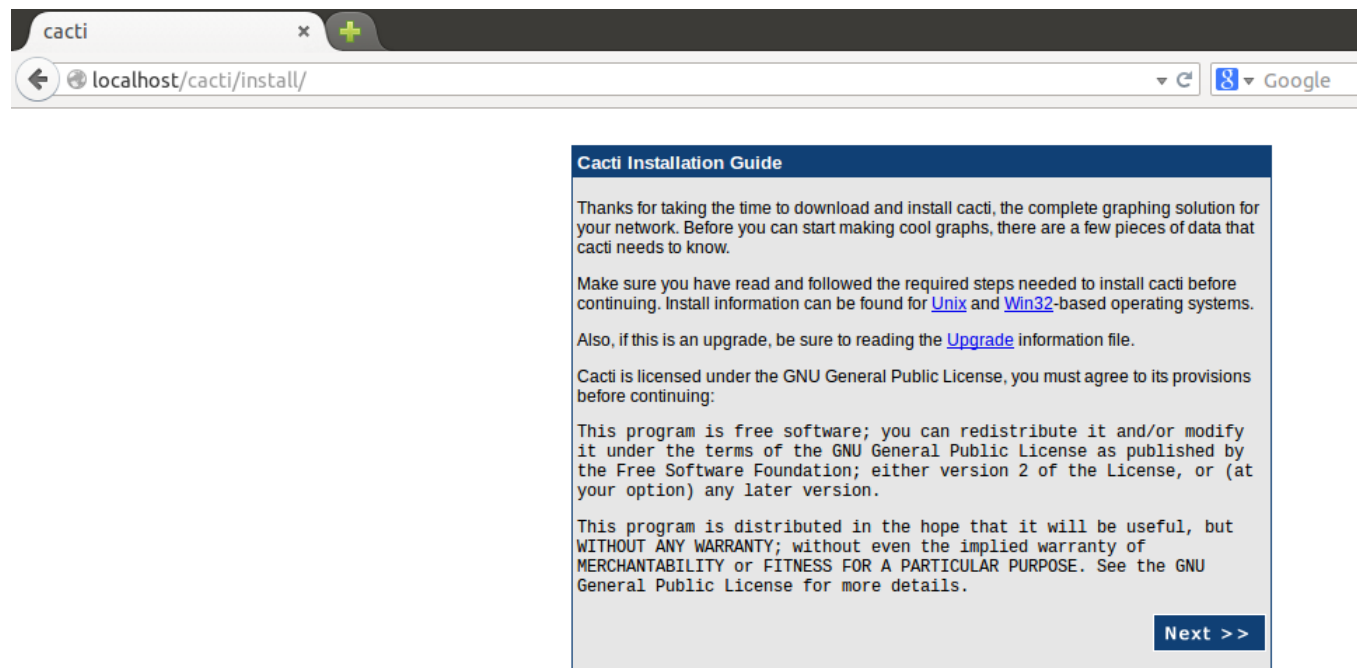


Figura 7.5: Instalación de cacti I

le damos a next y ahora indicamos el tipo de instalación (figura 7.6) que queremos realizar, nosotros le daremos a la opción "new installation" en lugar de la actualizar a la última versión.

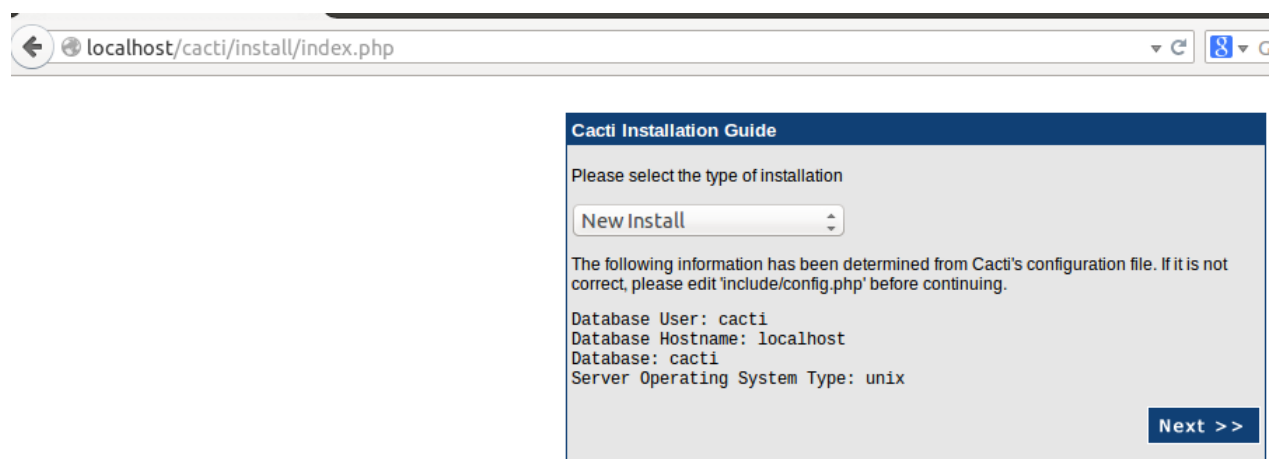


Figura 7.6: Instalación de cacti II

Lo siguiente que debemos realizar es poner cada fichero necesario en nuestro sistema, si algo no estuviese encontrado (found) tendríamos que buscarlo en nuestro sistema. En

la figura 7.7 podemos ver como todos los ficheros están correctos, por lo que pasamos al siguiente paso.

**Cacti Installation Guide**

Make sure all of these values are correct before continuing.

**[FOUND] RRDTool Binary Path:** The path to the rrdtool binary.  
/usr/bin/rrdtool  
**[OK: FILE FOUND]**

**[FOUND] PHP Binary Path:** The path to your PHP binary file (may require a php recompile to get this file).  
/usr/bin/php  
**[OK: FILE FOUND]**

**[FOUND] snmpwalk Binary Path:** The path to your snmpwalk binary.  
/usr/bin/snmpwalk  
**[OK: FILE FOUND]**

**[FOUND] snmpget Binary Path:** The path to your snmpget binary.  
/usr/bin/snmpget  
**[OK: FILE FOUND]**

**[FOUND] snmpbulkwalk Binary Path:** The path to your snmpbulkwalk binary.  
/usr/bin/snmpbulkwalk  
**[OK: FILE FOUND]**

**[FOUND] snmpgetnext Binary Path:** The path to your snmpgetnext binary.  
/usr/bin/snmpgetnext  
**[OK: FILE FOUND]**

**[FOUND] Cacti Log File Path:** The path to your Cacti log file.  
/var/log/cacti/cacti.log  
**[OK: FILE FOUND]**

**SNMP Utility Version:** The type of SNMP you have installed. Required if you are using SNMP v2c or don't have embedded SNMP support in PHP.  
NET-SNMP 5.x

**RRDTool Utility Version:** The version of RRDTool that you have installed.  
RRDTool 1.4.x

**NOTE:** Once you click "Finish", all of your settings will be saved and your database will be upgraded if this is an upgrade. You can change any of the settings on this screen at a later time by going to "Cacti Settings" from within Cacti.

**Finish**

Figura 7.7: Instalación de cacti III

Llegados a este punto nos debería salir lo que muestra la figura 7.8, en el que podemos loguearnos en la aplicación. En este caso lo haremos con el usuario admin y contraseña admin también, nada mas entrar nos pedirá cambiar la contraseña por una nueva.

**User Login**

Please enter your Cacti user name and password below:

User Name:

Password:

**Login**

Figura 7.8: Interfaz de acceso a cacti

En la figura 7.9 podemos ver la interfaz de cacti nada mas entrar.

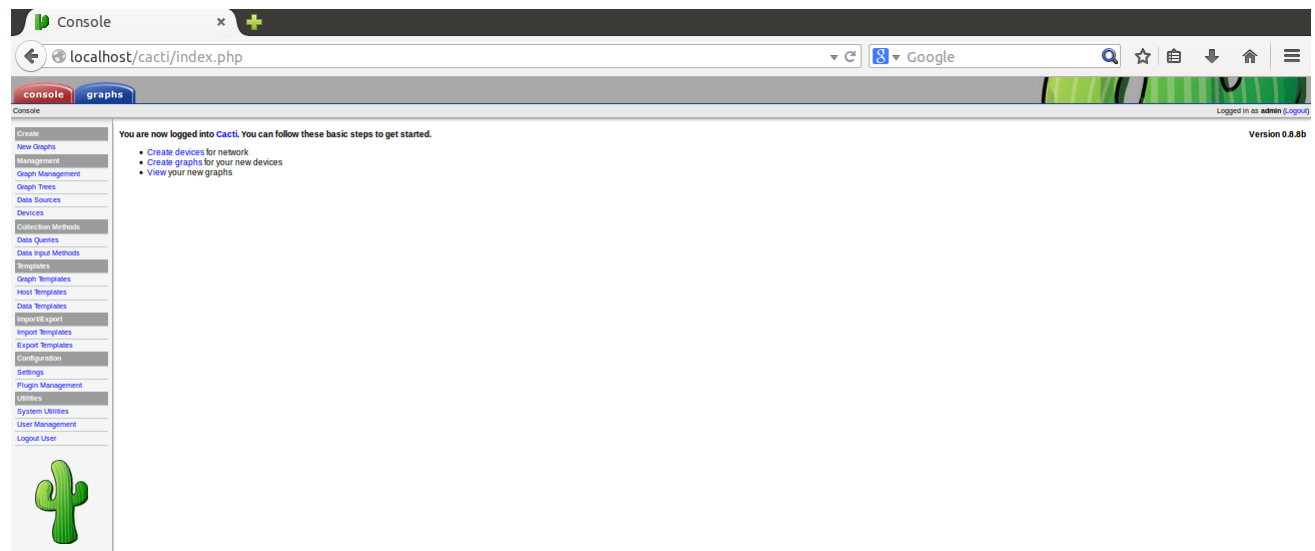


Figura 7.9: Interfaz de cacti

Por último realizamos una ejecución de nuestro monitor cacti, y aunque los tiempos de duración son muy pequeños podemos ver los datos de la gráfica, como muestra la figura 7.10

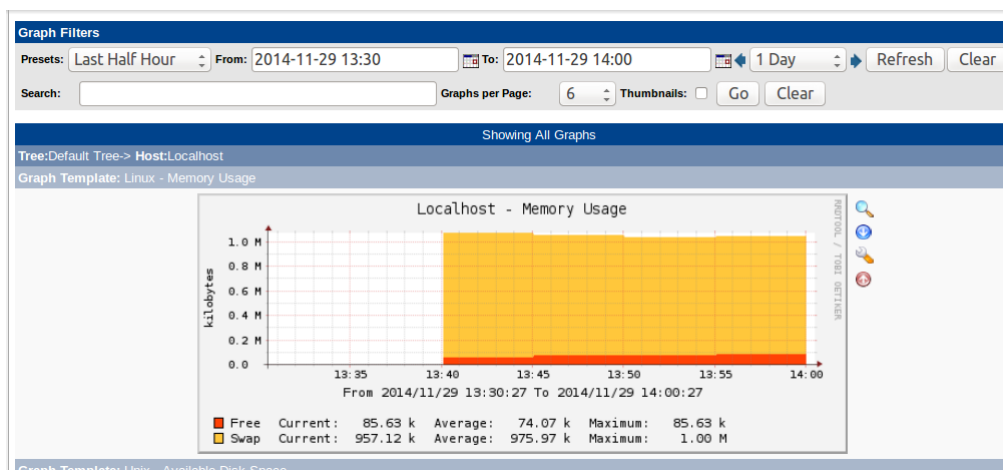


Figura 7.10: Ejemplo de ejecución de cacti I

podemos hacer que el monitor nos muestre mas información como en las figuras 7.11 y 7.12

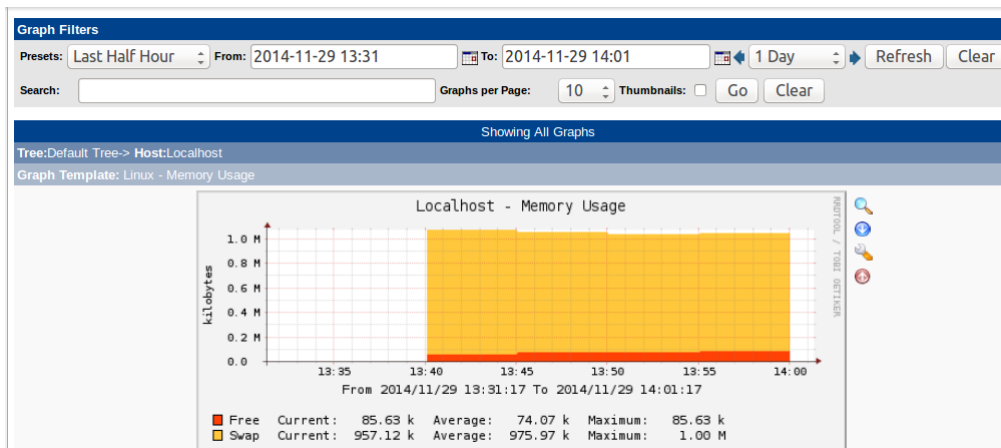


Figura 7.11: Monitorización de la memoria usada tanto libre como de intercambio

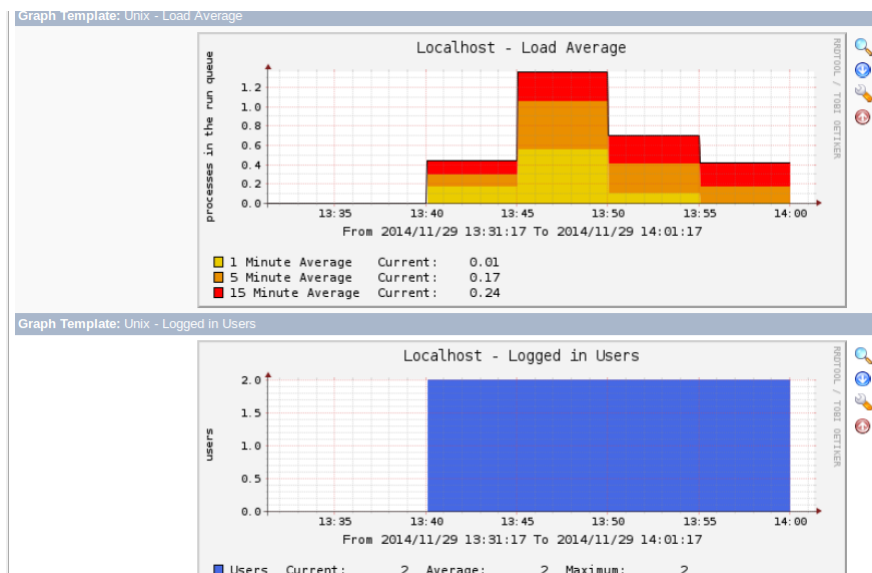


Figura 7.12: Muestras con la media de carga y los usuarios que han logueado

Además de estas se muestran muchas más gráficas que proporcionan mas información.

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

Strace [2] es una herramienta que se utiliza en la línea de comandos de linux y que te permite rastrear las llamadas al sistema y señales recibidas por un procesos y sus hijos. Strace puede usarse para arrancar directamente un programa o podemos conectarlo a él de manera que registre cada una de las llamadas al sistema que realice. Con strace podemos usar una gran variedad de flags que nos permitirán rastrear nuestros programas. Algunas de estos flags son -s que indica el tamaño máximo de la cadena que queremos imprimir y -f indica a strace que siga a los procesos hijo del proceso en cuestión, con la bandera -e podemos aceptar expresiones que le permite filtrar las trazas o cambiar como se han hecho dichas trazas, entre otras (para conocer el funcionamiento de más flags de strace solo tenemos que consultar el manual del sistema **man strace**). Strace se suele usar cuando se intenta solucionar un proceso sobre el que no se tiene mucho control, de manera que usando strace, Isot y las página del manual del sistema, podremos entender los detalles de lo que un programa está haciendo en su interior durante su ejecución.

## 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)

Creando en phpmyadmin una tabla llamada 'compra' con una tabla 'clientes' con dos columnas un 'id' que hace de clave primaria y un nombre a esta tabla le he introducido 3 tuplas.

Realizamos la siguiente consulta[3]:

**SELECT \* FROM clientes LIMIT 0 , 30**

El resultado lo perfilamos y nos queda como aparece en la figura 9.1.

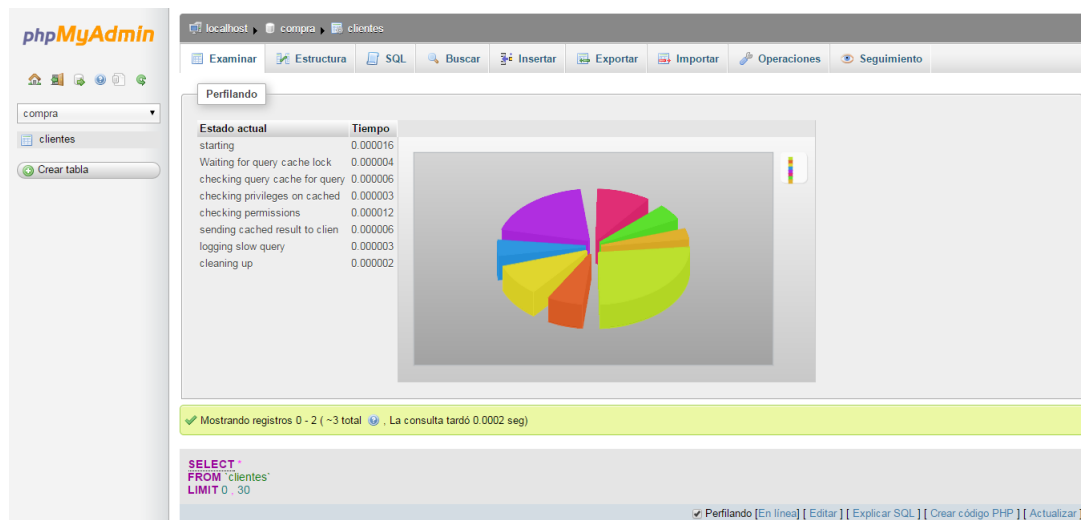


Figura 9.1: Perfil de una consulta sencilla en phpmyadmin I

Como vemos en la figura anterior al perfilar una consulta nos ofrece información acerca del tiempo que tardó en ejecutarse (en este caso al ser una consulta muy sencilla gasta muy poco tiempo 0.0002 seg ). Vemos que donde más tiempo gasta es en comenzar (starting) y en comprobar los permisos (checking permissions).

Si realizamos otra consulta sobre otra base de datos y otra tabla distinta podemos ver en la figura 9.2 como dichos valores y el tiempo de ejecución cambian.

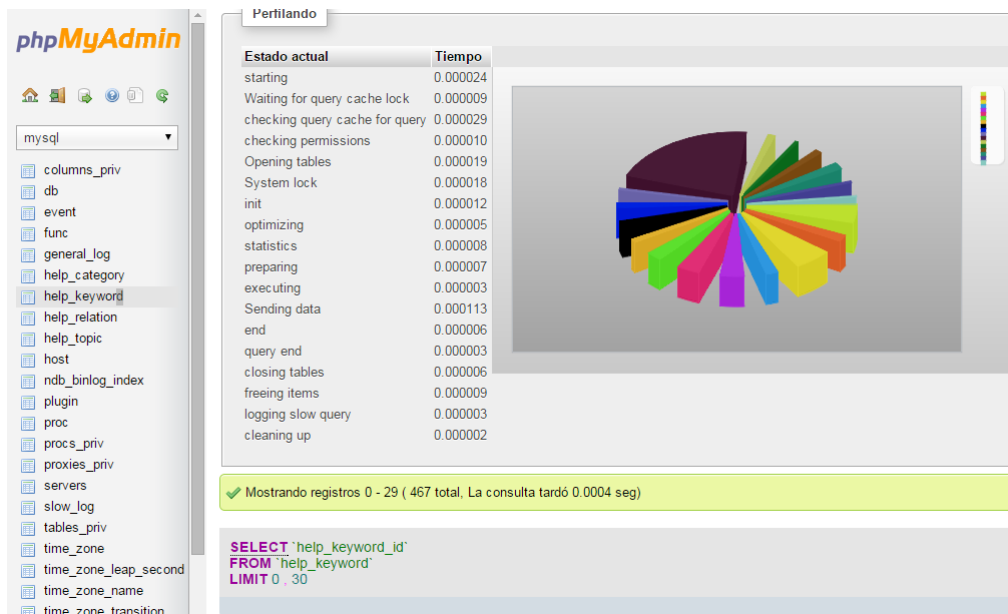


Figura 9.2: Perfil de una consulta sencilla en phpmyadmin II

## Referencias

- [1] Apuntes de la asignatura de sistemas operativos.
- [2] <http://chadfowler.com/blog/2014/01/26/the-magic-of-strace/>.
- [3] <http://kb.iweb.com/entries/71426693-ejecutar-consultas-sql-en-una-base-de-datos-con-phpmyadmin?locale=2>.
- [4] <http://www.alcpu.com/coretemp/>.
- [5] <http://www.cpubid.com/software/hwmonitor.html#top>.
- [6] <http://www.estrellateyarde.org/logs-en-linux>.
- [7] <http://www.ubuntugeek.com/how-to-install-cacti-monitoring-tool-in-ubuntu-13-10-server.html>.
- [8] Página 5 del manual de crontab: <http://unixhelp.ed.ac.uk/cgi/man-cgi?crontab+5>.
- [9] Página del manual, comando dmesg.