

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

Memoria Prácticas Opcionales

Jose Antonio Jiménez Montañés

16 de enero de 2015

Índice

1. Muestre (con capturas de pantalla) como ha comprobado que el RAID1 funciona	5
2. ¿Que relación hay entre los atajos de teclado de emacs y los de la consola bash?.¿y entre los de vi y las paginas del manual?	5
3. ¿Que gestores utiliza OpenSuse?	5
4. Instale y pruebe terminator. Con screen, pruebe su funcionamiento dejando sesiones ssh abiertas en el servidor y recuperándolas posteriormente.	6
5. Instale el servicio fail2ban y pruebe su funcionamiento.	7
6. Realice la instalación de uno de estos dos “web containers” y pruebe su ejecución.	8
7. Realice la instalación de MongoDB en alguna de sus máquinas virtuales. Cree una colección de documentos y haga una consulta sobre ellos.	9
8. Indique qué comandos ha utilizado para realizarlo así como capturas de pantalla del proceso de reconstrucción del RAID. [1]	10
9. Instale Nagios en su sistema (el que prefiera) documentando el proceso y muestre el resultado de la monitorización de su sistema comentando qué aparece. [5]	11
10. Con Ganglia haga lo mismo que con Munin.	16
11. Prueba a instalar este monitor en alguno de sus tres sistemas. Realice capturas de pantalla del proceso de instalación y comente capturas de pantalla del programa en ejecución.[8]	16
12. 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	18
13. Desarrolle una página en C o C++ y analice su comportamiento usando valgrind.	25
14. Seleccione, instale y ejecute uno, comente los resultados.	29
15. ¿Qué es Scala? Instale Gatling y pruebe los escenarios por defecto. [6]	31
16. Lea el artículo sobre Jmeter y elabore un breve resumen.	32

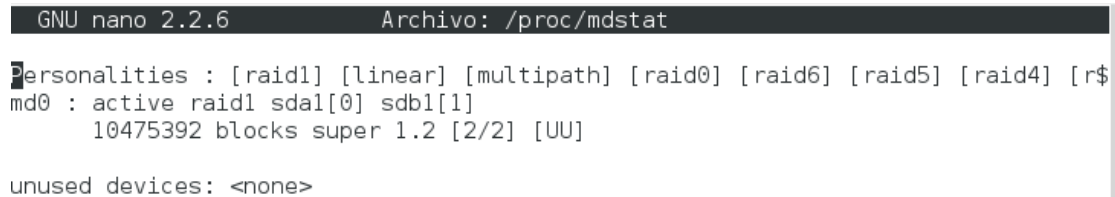
17. Seleccione un benchmark entre SisoftSandra y Aida. Ejecútelo y muestre capturas de pantalla comentando los resultados. **32**

Índice de figuras

1.1. Comprobación del estado del RAID1	5
4.1. Dependencias faltantes para instalar rpm de terminator	6
4.2. Uso del comando screen	6
5.1. Instalación completada de fail2ban	7
5.2. Servicio fail2ban funcionando	7
6.1. Instalación de Java completada	8
6.2. Instalación de TomCat completada	8
6.3. Instalación de TomCat probada	8
7.1. Instalación de MongoDB completada	9
7.2. Proceso de creación y consulta de la colección de datos	9
8.1. Sistema detecta que el RAID falla	10
8.2. Sistema detecta nuevo disco en RAID y lo actualiza	10
9.1. Prerequisitos de nagios con apt-get.	11
9.2. Creacion de usuario y grupo Nagios.	12
9.3. Archivos de instalación de Nagios descomprimidos.	12
9.4. Preconfiguración de Nagios completada.	13
9.5. Instalacion de Nagios completada y servicio corriendo.	14
9.6. Home Web de Nagios funcionando.	15
10.1. Monitorización de servidor de pruebas con Munin	16
11.1. Pantalla de login de Zabbix.	16
11.2. Pantalla Principal de Zabbix.	17
12.1. Pantalla inicial de preinstalación de Cacti.	18
12.2. Opción de nueva instalación o actualización.	19
12.3. Comprobación de rutas en la instalación de cacti.	20
12.4. Login en cacti.	21
12.5. Cambio de password para admin en cacti.	22
12.6. Home de cacti.	23
12.7. Gráficas de muestreo en cacti.	24
13.1. Resultado en consola de valgring sobre un programa de multiplicación de matrices.	28
14.1. Instalacion del benchmark hdparm-read	29
14.2. Ejecución del benchmark hdparm-read	29
14.3. Resumen de nuestro sistema y confirmación de guardado de datos.	30
14.4. Realización de los test de disco.	30
14.5. Resultado del benchmark hdparm-read.	31
14.6. Resultado del benchmark hdparm-read en gráfica.	31
17.1. Pantalla inicial de SisoftSandra.	32
17.2. Diversos test de SisoftSandra.	33
17.3. Resultado de testear la CPU con SiSoftSandra.	33

1. Muestre (con capturas de pantalla) como ha comprobado que el RAID1 funciona

Consultando el archivo /proc/mdstat. Si los 2 discos RAID están funcionando correctamente el sistema raid se marcará con una doble U ('UU') si alguno fallase mostraría solo una U con una barra baja en el lugar del disco con fallo ('_U').

A screenshot of a terminal window showing the contents of the /proc/mdstat file. The window title is 'GNU nano 2.2.6 Archivo: /proc/mdstat'. The output shows the RAID configuration: 'Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [r\$' and 'md0 : active raid1 sda1[0] sdb1[1] 10475392 blocks super 1.2 [2/2] [UU]'. At the bottom, it says 'unused devices: <none>'.

```
GNU nano 2.2.6 Archivo: /proc/mdstat
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [r$
md0 : active raid1 sda1[0] sdb1[1]
      10475392 blocks super 1.2 [2/2] [UU]

unused devices: <none>
```

Figura 1.1: Comprobación del estado del RAID1

2. ¿Que relación hay entre los atajos de teclado de emacs y los de la consola bash?.¿y entre los de vi y las paginas del manual?

Los atajos de emacs son casi idénticos a los de bash ya que es desarrollado por GNU aunque se puede cambiar la configuración al estilo Vi/Vim escribiendo: 'set -o vi'.

Los atajos de las páginas del manual son idénticos a los del vi aunque internamente parece que no tienen que ver ya que las "man pages" no se basan en vi sino en "boost".

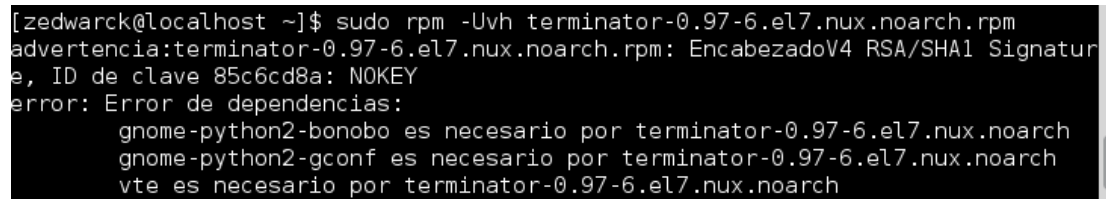
3. ¿Que gestores utiliza OpenSuse?

De manera gráfica utiliza YaST, y para el modo consola utiliza Zypper.

4. Instale y pruebe terminator. Con screen, pruebe su funcionamiento dejando sesiones ssh abiertas en el servidor y recuperándolas posteriormente.

Primero ya que estoy trabajando con Centos he tenido que descargar el rpm con terminator de: http://li.nux.ro/download/nux/dextop/el7/x86_64/terminator-0.97-6.el7.nux.noarch.rpm

Luego he instalado las 3 dependencias que son necesarias para instalar el rpm de terminator que son:



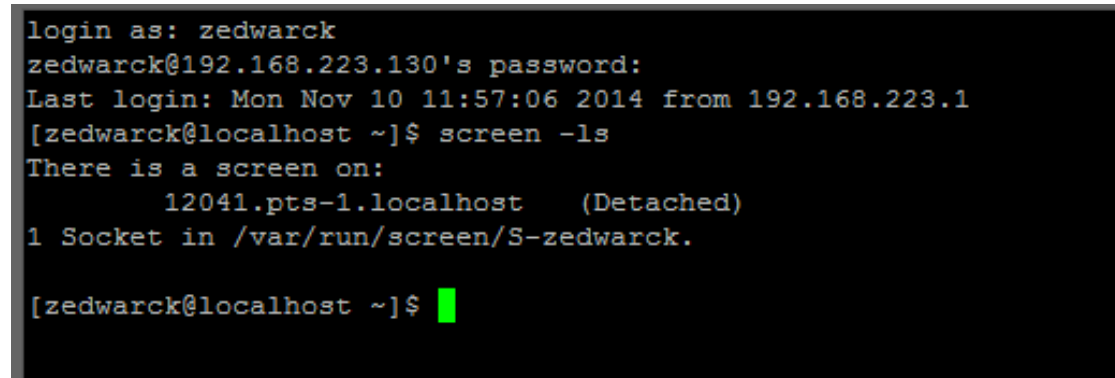
```
[zedwarck@localhost ~]$ sudo rpm -Uvh terminator-0.97-6.el7.nux.noarch.rpm
advertencia:terminator-0.97-6.el7.nux.noarch.rpm: EncabezadoV4 RSA/SHA1 Signature, ID de clave 85c6cd8a: NOKEY
error: Error de dependencias:
    gnome-python2-bonobo es necesario por terminator-0.97-6.el7.nux.noarch
    gnome-python2-gconf es necesario por terminator-0.97-6.el7.nux.noarch
    vte es necesario por terminator-0.97-6.el7.nux.noarch
```

Figura 4.1: Dependencias faltantes para instalar rpm de terminator

Finalmente se instala el comando screen con: `yum install screen`. NOTA: Puede que se necesite tener los repositorios EPEL instalados. [7]

Ya se puede acceder desde la sesión SSH al comando screen para que nos virtualice una sesión de terminal que no se cerrara en caso de desconexión del túnel ssh.

Si se desconecta se puede ver las sesiones virtuales abiertas con `screen -ls` y acceder a una de ellas usando su ID con `screen -r <ID>`[10]



```
login as: zedwarck
zedwarck@192.168.223.130's password:
Last login: Mon Nov 10 11:57:06 2014 from 192.168.223.1
[zedwarck@localhost ~]$ screen -ls
There is a screen on:
    12041.pts-1.localhost    (Detached)
1 Socket in /var/run/screen/S-zedwarck.

[zedwarck@localhost ~]$
```

Figura 4.2: Uso del comando screen

5. Instale el servicio fail2ban y pruebe su funcionamiento.

Primero instalamos el paquete con yum install fail2ban

```
Instalado:
  fail2ban.noarch 0:0.9-9.el7

Dependencia(s) instalada(s):
  fail2ban-firewalld.noarch 0:0.9-9.el7    fail2ban-sendmail.noarch 0:0.9-9.el7
  fail2ban-server.noarch 0:0.9-9.el7       fail2ban-systemd.noarch 0:0.9-9.el7
  ipset.x86_64 0:6.19-4.el7               ipset-libs.x86_64 0:6.19-4.el7

¡Listo!
[zedwarck@localhost ~]$
```

Figura 5.1: Instalación completada de fail2ban

Y activamos el servicio y comprobamos que esta ejecutandose:

```
[zedwarck@localhost ~]$ service fail2ban start
Redirecting to /bin/systemctl start fail2ban.service
Failed to issue method call: Access denied
[zedwarck@localhost ~]$ sudo service fail2ban start
[sudo] password for zedwarck:
Redirecting to /bin/systemctl start fail2ban.service
[zedwarck@localhost ~]$ service fail2ban status
Redirecting to /bin/systemctl status fail2ban.service
fail2ban.service - Fail2ban Service
   Loaded: loaded (/usr/lib/systemd/system/fail2ban.service; disabled)
   Active: active (running) since lun 2014-11-10 20:27:34 CET; 9s ago
     Process: 20178 ExecStart=/usr/bin/fail2ban-client -x start (code=exited, status=0/SUCCESS)
    Main PID: 20181 (fail2ban-server)
      CGroup: /system.slice/fail2ban.service
              └─20181 /usr/bin/python /usr/bin/fail2ban-server -b -s /var/run/fa...

[zedwarck@localhost ~]$
```

Figura 5.2: Servicio fail2ban funcionando

Después solo tendremos que crear el archivo de configuración y configurarlo a nuestro gusto para restringir ciertas IPS, incluso con aviso por email en caso de intento no autorizado. [9]

6. Realice la instalación de uno de estos dos “web containers” y pruebe su ejecución.

Primero instalamos Java

```
Actualizado:
 java-1.7.0-openjdk.x86_64 1:1.7.0.71-2.5.3.1.el7_0

Dependencia(s) actualizada(s):
 java-1.7.0-openjdk-headless.x86_64 1:1.7.0.71-2.5.3.1.el7_0

¡Listo!
[zedwarck@localhost ~]$
```

Figura 6.1: Instalación de Java completada

Luego TomCat

```
Instalado:
 tomcat.noarch 0:7.0.42-8.el7_0
 tomcat-el-2.2-api.noarch 0:7.0.42-8.el7_0
 tomcat-jsvc.noarch 0:7.0.42-8.el7_0
 tomcat-servlet-3.0-api.noarch 0:7.0.42-8.el7_0
 tomcat-admin-webapps.noarch 0:7.0.42-8.el7_0
 tomcat-javadoc.noarch 0:7.0.42-8.el7_0
 tomcat-lib.noarch 0:7.0.42-8.el7_0
 tomcat-webapps.noarch 0:7.0.42-8.el7_0
 tomcat-docs-webapp.noarch 0:7.0.42-8.el7_0
 tomcat-jsp-2.2-api.noarch 0:7.0.42-8.el7_0
 tomcat-native.x86_64 0:1.1.30-1.el7

Dependencia(s) instalada(s):
 apache-commons-collections.noarch 0:3.2.1-21.el7
 apache-commons-daemon-jsvc.x86_64 0:1.0.13-6.el7
 apache-commons-logging.noarch 0:1.1.2-7.el7
 avalon-framework.noarch 0:4.3-10.el7
 ecj.x86_64 1:4.2.1-8.el7
 geronimo-jta.noarch 0:1.1.1-17.el7
 javamail.noarch 0:1.4.6-8.el7
 xalan-j2.noarch 0:2.7.1-23.el7
 xml-commons-apis.noarch 0:1.4.01-16.el7
 apache-commons-daemon.x86_64 0:1.0.13-6.el7
 apache-commons-dbc.noarch 0:1.4-17.el7
 apache-commons-pool.noarch 0:1.6-9.el7
 avalon-logkit.noarch 0:2.1-14.el7
 geronimo-jms.noarch 0:1.1.1-19.el7
 jakarta-taglibs-standard.noarch 0:1.1.2-11.el7
 log4j.noarch 0:1.2.17-15.el7
 xerces-j2.noarch 0:2.11.0-17.el7_0
 xml-commons-resolver.noarch 0:1.2-15.el7

Ignorando (problemas de dependencias):
 jss.x86_64 0:4.2.6-33.el7
 tomcatjss.noarch 0:7.1.0-4.el7

¡Listo!
[zedwarck@localhost ~]$
```

Figura 6.2: Instalación de TomCat completada

Luego configuramos Tomcat [2] y probamos poniendo la ip local en el puerto 8080 en el navegador.

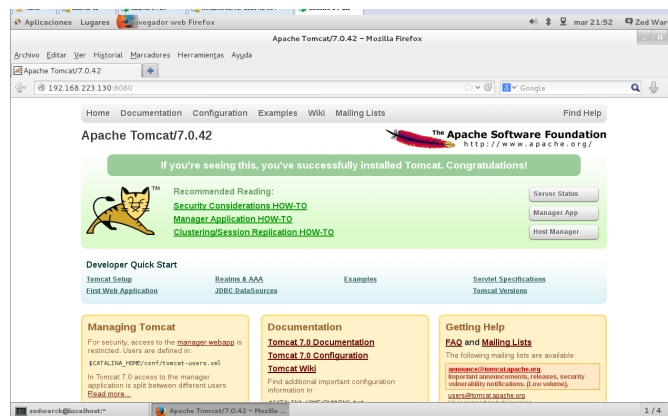


Figura 6.3: Instalación de TomCat probada

7. Realice la instalación de MongoDB en alguna de sus máquinas virtuales. Cree una colección de documentos y haga una consulta sobre ellos.

Primero instalamos MongoDB y lo configuramos: [3]

```
Instalado:
  mongodb-org.x86_64 0:2.6.5-1

Dependencia(s) instalada(s):
  mongodb-org-mongos.x86_64 0:2.6.5-1   mongodb-org-server.x86_64 0:2.6.5-1
  mongodb-org-shell.x86_64 0:2.6.5-1   mongodb-org-tools.x86_64 0:2.6.5-1

¡Listo!
[zedwarck@localhost yum.repos.d]$
```

Figura 7.1: Instalación de MongoDB completada

Y luego nos conectamos y creamos las colecciones de prueba: [4]

```
Instalado:
  mongodb-org.x86_64 0:2.6.5-1

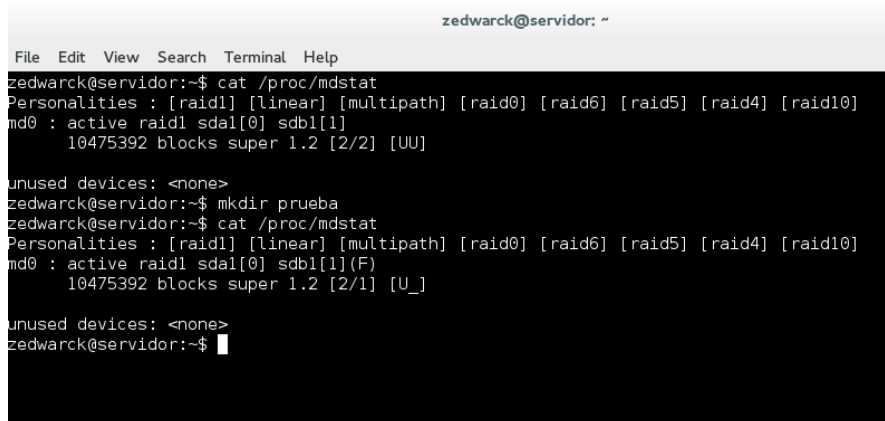
Dependencia(s) instalada(s):
  mongodb-org-mongos.x86_64 0:2.6.5-1   mongodb-org-server.x86_64 0:2.6.5-1
  mongodb-org-shell.x86_64 0:2.6.5-1   mongodb-org-tools.x86_64 0:2.6.5-1

¡Listo!
[zedwarck@localhost yum.repos.d]$
```

Figura 7.2: Proceso de creación y consulta de la colección de datos

8. Indique qué comandos ha utilizado para realizarlo así como capturas de pantalla del proceso de reconstrucción del RAID. [1]

Comprobamos primero que funciona el RAID1 y eliminamos el disco2. Comprobamos que detecta que el disco 2 falla: (Lo hemos eliminado)



```
zedwarck@servidor: ~  
File Edit View Search Terminal Help  
zedwarck@servidor:~$ cat /proc/mdstat  
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]  
md0 : active raid1 sdal[0] sdb1[1]  
      10475392 blocks super 1.2 [2/2] [UU]  
  
unused devices: <none>  
zedwarck@servidor:~$ mkdir prueba  
zedwarck@servidor:~$ cat /proc/mdstat  
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]  
md0 : active raid1 sdal[0] sdb1[1](F)  
      10475392 blocks super 1.2 [2/1] [U_]   
  
unused devices: <none>  
zedwarck@servidor:~$
```

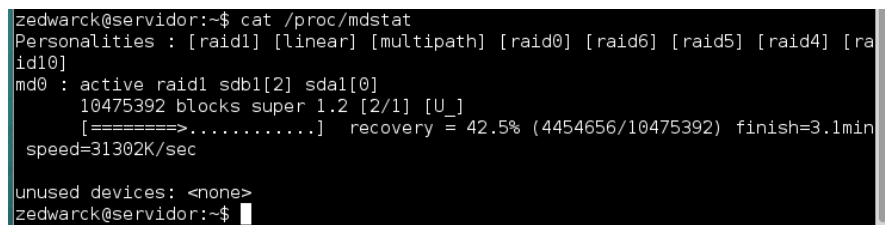
Figura 8.1: Sistema detecta que el RAID falla

Si el disco diera fallos lógicos debería ser retirado con: "sudo mdadm --remove /dev/md0 /dev/sdb1"

De una forma u de otra el siguiente paso sería retirarlo del sistema con: "sudo mdadm --manage /dev/md0 --remove /dev/sdb1"

Luego procedemos a instalar el nuevo disco que ya debería estar en el sistema con: "sudo mdadm --add /dev/md0 /dev/sdb1"

Comprobamos que ahora esta el disco en el sistema RAID y esta actualizando datos:



```
zedwarck@servidor:~$ cat /proc/mdstat  
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]  
md0 : active raid1 sdb1[2] sdal[0]  
      10475392 blocks super 1.2 [2/1] [U_]   
      [=====>.....]  recovery = 42.5% (4454656/10475392) finish=3.1min  
      speed=31302K/sec  
  
unused devices: <none>  
zedwarck@servidor:~$
```

Figura 8.2: Sistema detecta nuevo disco en RAID y lo actualiza

9. Instale Nagios en su sistema (el que prefiera) documentando el proceso y muestre el resultado de la monitorización de su sistema comentando qué aparece. [5]

Para instalar Nagios en Ubuntu primero preinstalamos algunos requisitos. segun el manual habría que poner:

```
sudo apt-get install wget build-essential apache2 php5-gd libgd2-xpm libgd2-xpm-dev libapache2-modphp5
```

Pero vamos a dejar la linea de la siguiente manera para Ubuntu 14.04:

```
sudo apt-get install wget build-essential apache2 php5-gd libgd2-xpm-dev apache2-utils
```



```
zedwarck@servidor: ~/Downloads
File Edit View Search Terminal Help
zedwarck@servidor:~/Downloads$ ls
nagios-4.0.8.tar.gz
zedwarck@servidor:~/Downloads$ sudo apt-get install wget build-essential apache2 php5-gd libgd2-xpm-dev libapache2-mod-php5
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
build-essential ya está en su versión más reciente.
libgd2-xpm-dev ya está en su versión más reciente.
apache2 ya está en su versión más reciente.
libapache2-mod-php5 ya está en su versión más reciente.
php5-gd ya está en su versión más reciente.
fijado php5-gd como instalado manualmente.
wget ya está en su versión más reciente.
0 actualizados, 0 se instalarán, 0 para eliminar y 0 no actualizados.
zedwarck@servidor:~/Downloads$
```

Figura 9.1: Prerequisitos de nagios con apt-get.

Luego nos descargamos nagios:

```
cd /tmp
```

```
wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.4.tar.gz
```

```
wget http://nagios-plugins.org/download/nagios-plugins-2.0.tar.gz
```

Nos creamos un grupo y un usuario en él que se necesita para su correcto funcionamiento:

```
useradd nagios
```

```
groupadd nagcmd
```

```
usermod -a -G nagcmd nagios
```

```
root@servidor: /home/zedwarck/Downloads
File Edit View Search Terminal Help
zedwarck@servidor:~/Downloads$ sudo su
root@servidor:/home/zedwarck/Downloads# useradd nagios
useradd: user 'nagios' already exists
root@servidor:/home/zedwarck/Downloads# groupadd nagcmd
groupadd: group 'nagcmd' already exists
root@servidor:/home/zedwarck/Downloads# usermod -a -G nagcmd nagios
root@servidor:/home/zedwarck/Downloads#
```

Figura 9.2: Creacion de usuario y grupo Nagios.

Descomprimimos:

`tar zxvf nagios-4.0.4.tar.gz`

`tar zxvf nagios-plugins-2.0.tar.gz`

```
root@servidor:/home/zedwarck/Downloads# ls
nagios-4.0.8  nagios-4.0.8.tar.gz  nagios-plugins-2.0.3  nagios-plugins-2.0.3.tar.gz
root@servidor:/home/zedwarck/Downloads#
```

Figura 9.3: Archivos de instalación de Nagios descomprimidos.

Accedemos al directorio de instalación y configuramos el instalador:

`./configure --with-nagios-group=nagios --with-command-group=nagcmd --with-mail=/usr/bin/sendmail`

```

config.status: creating lib/iobroker.h
Creating sample config files in sample-config/ ...

*** Configuration summary for nagios 4.0.8 08-12-2014 ***:

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

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

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

root@servidor:/home/zedwarck/Downloads/nagios-4.0.8# █

```

Figura 9.4: Preconfiguración de Nagios completada.

Luego hacemos los makes:

```

make all
make install
make install-init
make install-config
make install-commandmode
make install-webconf

```

```

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

Running pre-flight check on configuration data...

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

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
root@servidor:/home/zedwarck/Downloads/nagios-4.0.8# /etc/init.d/nagios start
Starting nagios: done.
root@servidor:/home/zedwarck/Downloads/nagios-4.0.8# █

```

Figura 9.5: Instalacion de Nagios completada y servicio corriendo.

El último puede que nos de error, si es así nos creamos el directorio `/etc/httpd/conf.d` y volvemos a ejecutar el ultimo make. Después de eso nos vamos a ese directorio y copiamos el archivo `nagios.conf` al directorio de configuraciones de apache: `/etc/apache2/conf-enabled/`

Finalmente ponemos:

```

cp -R contrib/eventhandlers/ /usr/local/nagios/libexec/
chown -R nagios:nagios /usr/local/nagios/libexec/eventhandlers
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

```

Iniciamos el servicio naglios: `service nagios restart`

Por ultimo reiniciamos apache: `service apache2 restart`

Para poder acceder nos hace falta configurar un usuario:

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

Y para poder visualizar gráficas nos hace falta el plugin:

```
cd /tmp/nagios-plugins-2.0
./configure --with-nagios-user=nagios --with-nagios-group=nagios
make
make install
```

Con esto ya casi lo tendríamos todo, solo nos faltaría configurar nuestro apache para que acepte scripts CGI.

Para acceder, desde un navegador ponemos la IP o localhost(si esta activo desde el archivo .conf)/nagios:

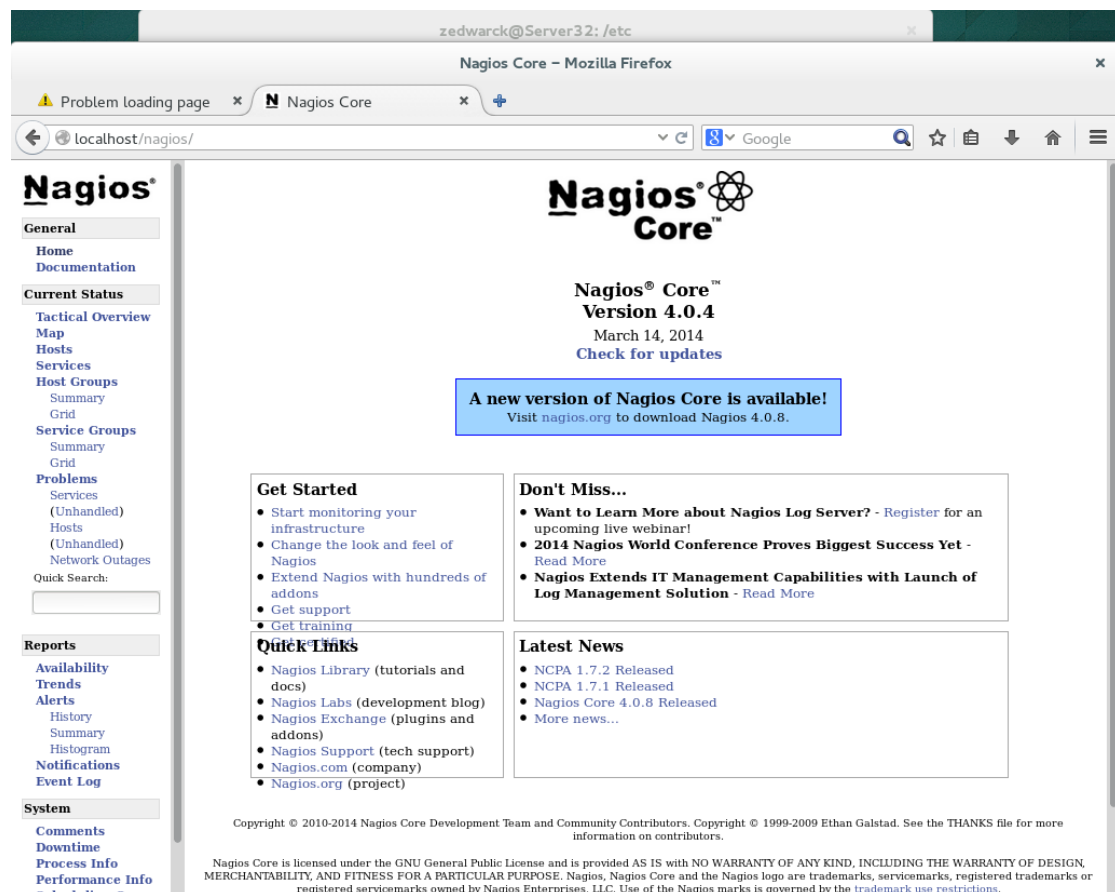


Figura 9.6: Home Web de Nagios funcionando.

10. Con Ganglia haga lo mismo que con Munin.

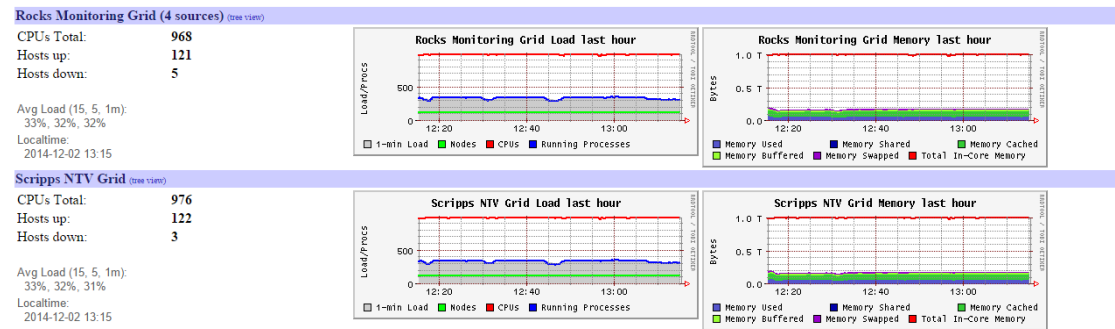


Figura 10.1: Monitorización de servidor de pruebas con Munin

11. Prueba a instalar este monitor es alguno de sus tres sistemas. Realice capturas de pantalla del proceso de instalación y comente capturas de pantalla del programa en ejecución.[8]

Instalamos y configuramos nuestro servidor con zabbix siguiendo los pasos indicados en la cita(exceptuando el usuario de la base de datos que en nuestro caso era root).

Una vez instalado podemos acceder con el navegador:



Figura 11.1: Pantalla de login de Zabbix.

Introducimos usuario: admin y clave: zabbix, para acceder y ya tendremos la pantalla de inicio:

Monitoring | Inventory | Reports | Configuration | Administration

Zabbix | Overview | Web | Latest data | Triggers | Events | Graphs | Screens | Maps | Discovery | IT services

History: Configuration of hosts » Configuration of items » Configuration of actions » Configuration of maintenance periods » Configuration of host groups

PERSONAL DASHBOARD

Favourite graphs

No graphs added.

[Graphs »](#)

Favourite screens

No screens added.

[Screens »](#)

Favourite maps

No maps added.

[Maps »](#)

Status of Zabbix

Parameter	Value	Details
Zabbix server is running	No	localhost:10051
Number of hosts (monitored/not monitored/templates)	39	1 / 0 / 38
Number of items (monitored/disabled/not supported)	63	63 / 0 / 0
Number of triggers (enabled/disabled) [problem/ok]	42	42 / 0 [0 / 42]
Number of users (online)	2	2
Required server performance, new values per second	0.94	-

Updated: 21:40:02

System status

Host group	Disaster	High	Average	Warning	Information	Not classified
Zabbix servers	0	0	0	0	0	0

Updated: 21:40:02

Host status

Host group	Without problems	With problems	Total
Zabbix servers	1	0	1

Updated: 21:41:02

Last 20 issues

Host	Issue	Last change	Age	Info	Ack	Actions
No events found.						

0 of 0 issues are shown

Updated: 21:41:02

Web monitoring

Host group	Ok	Failed	Unknown
No web scenarios found.			

Updated: 21:41:02

Figura 11.2: Pantalla Principal de Zabbix.

12. 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

Necesitamos tener apache, php5 y mysql como prerequisite, para ello ponemos:

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

Una vez tengamos todo los prerequisites, instalamos cacti:

```
sudo apt-get install cacti cacti-spine
```

Después de hacer toda la instalación ya ponemos ir a un navegador y poner como dirección: localhost/cacti

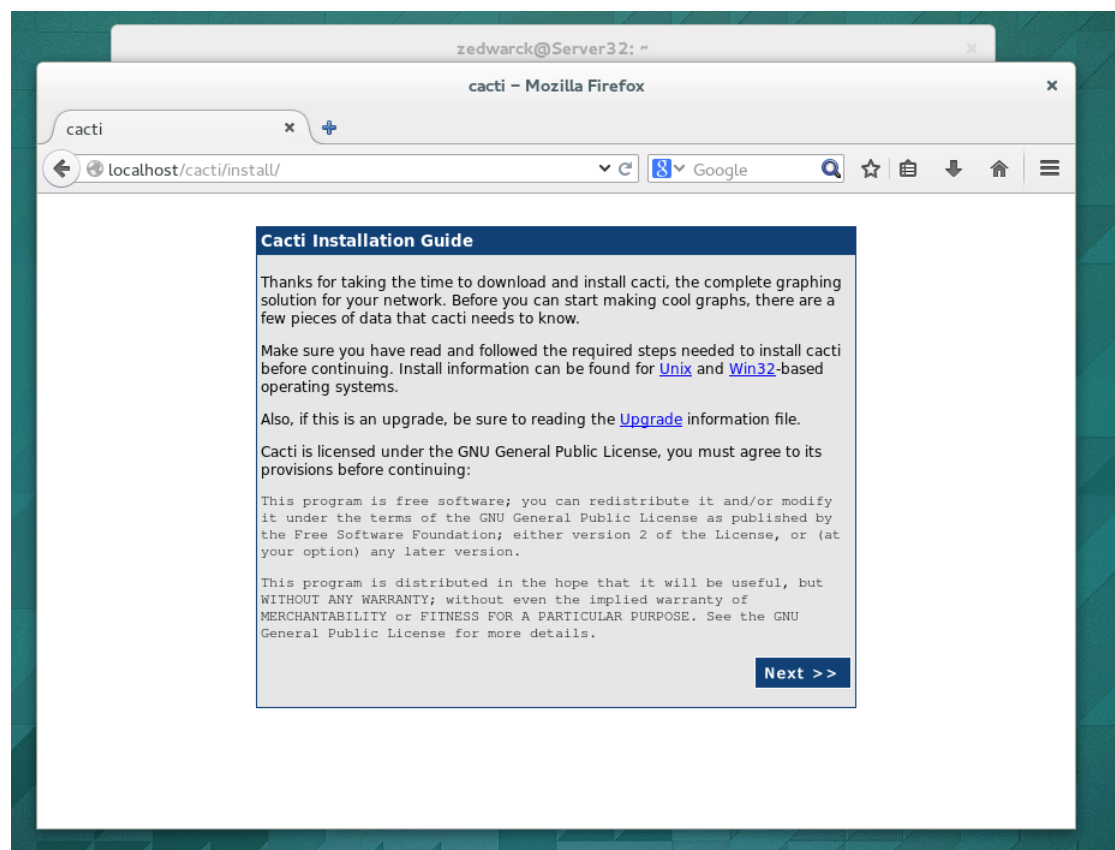


Figura 12.1: Pantalla inicial de preinstalación de Cacti.

Pulsamos a siguiente y le decimos que queremos una nueva instalación:

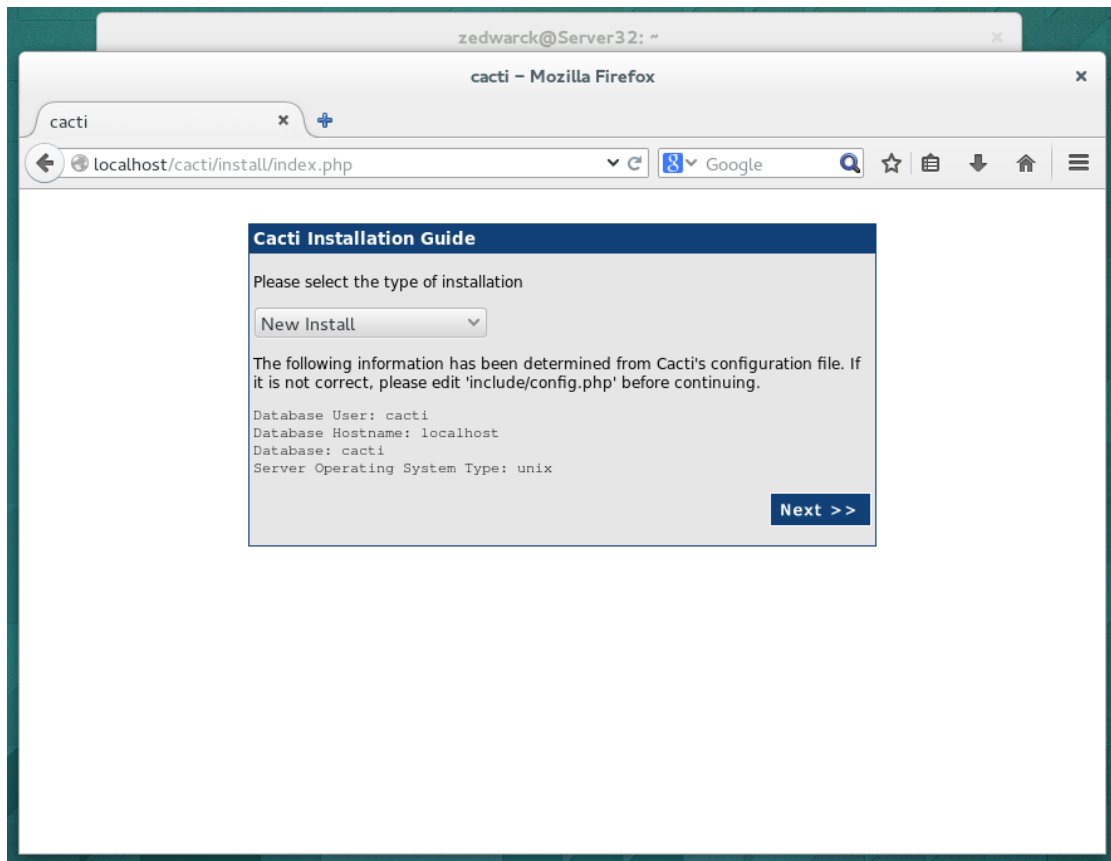


Figura 12.2: Opción de nueva instalación o actualización.

Luego el sistema comprueba que están los archivos que necesita en sus rutas adecuadas. Pulsamos terminar para completar el proceso:

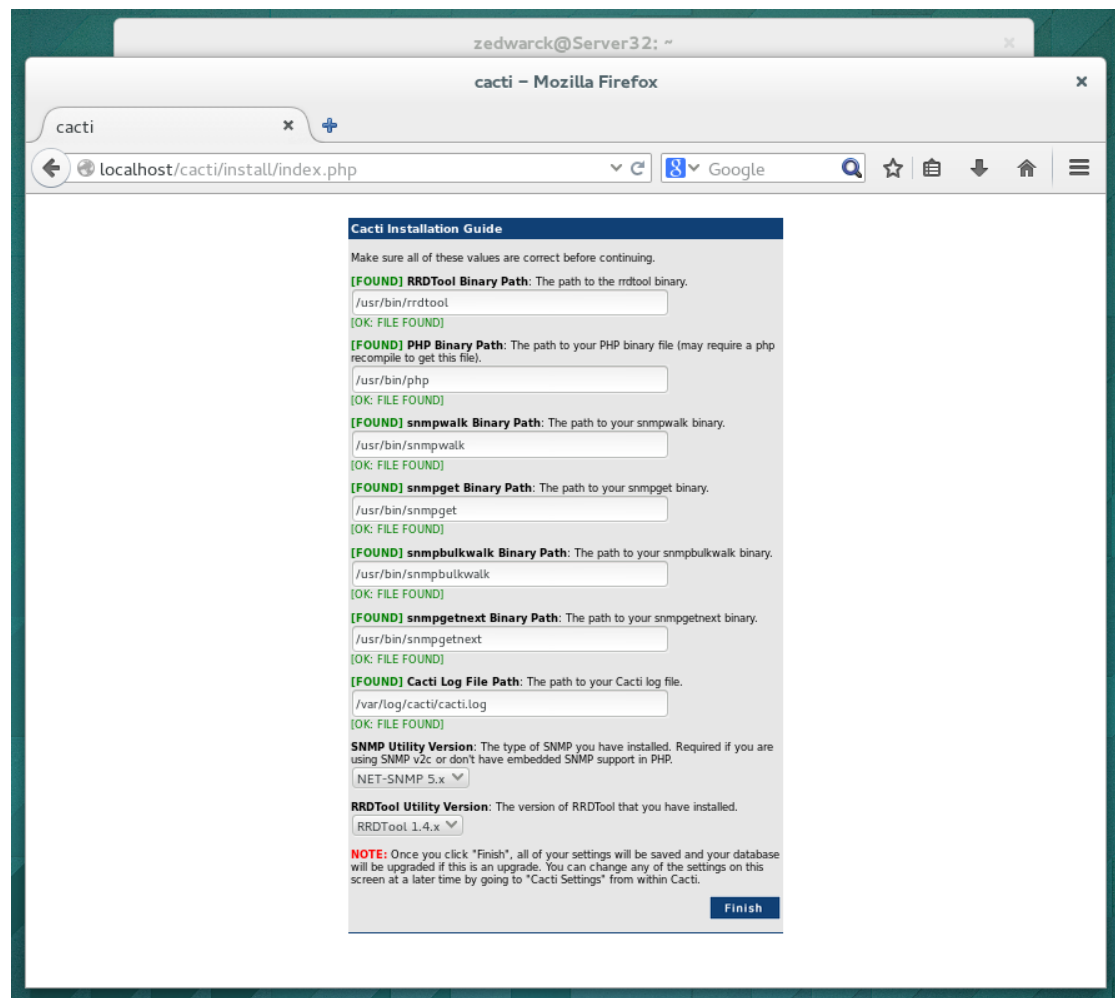


Figura 12.3: Comprobación de rutas en la instalación de cacti.

Luego nos pide un usuario y un password que le ponemos "admin admin":

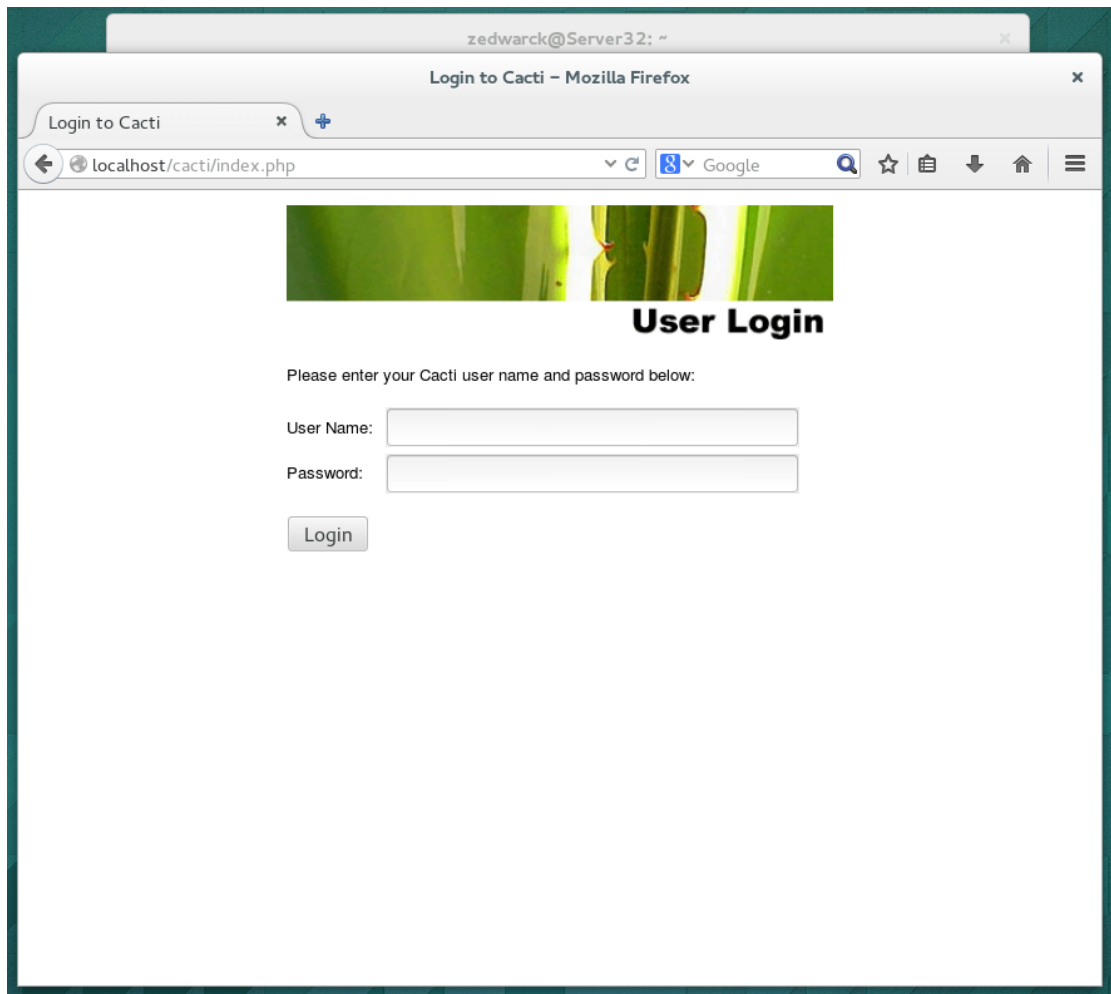


Figura 12.4: Login en cacti.

Luego nos obliga a cambiar de password:

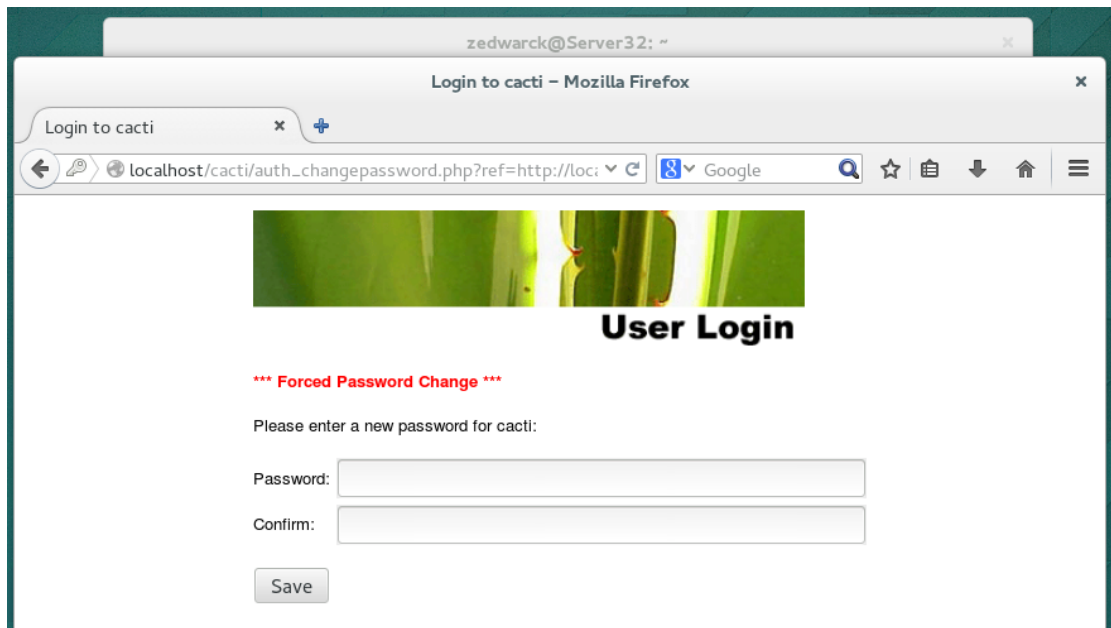


Figura 12.5: Cambio de password para admin en cacti.

Finalmente accedemos al entorno web de cacti:

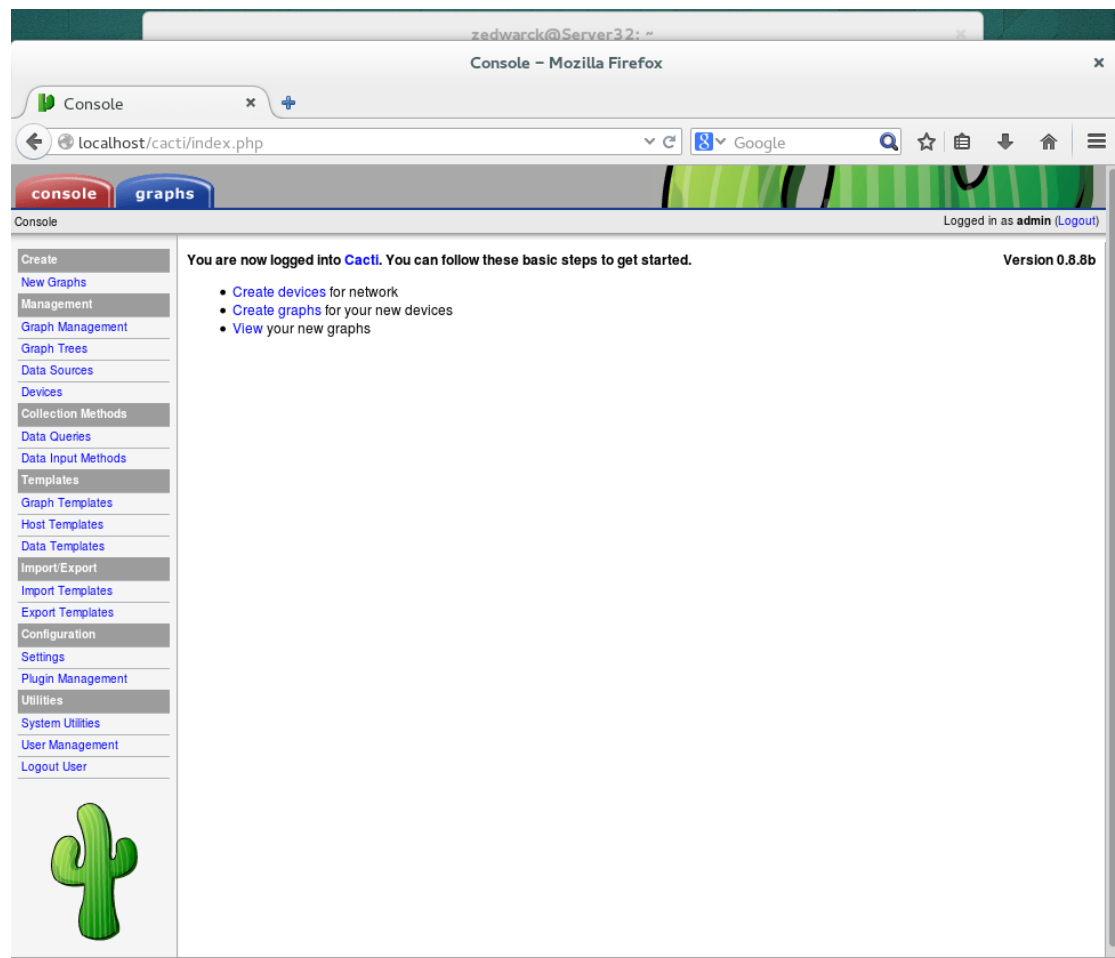
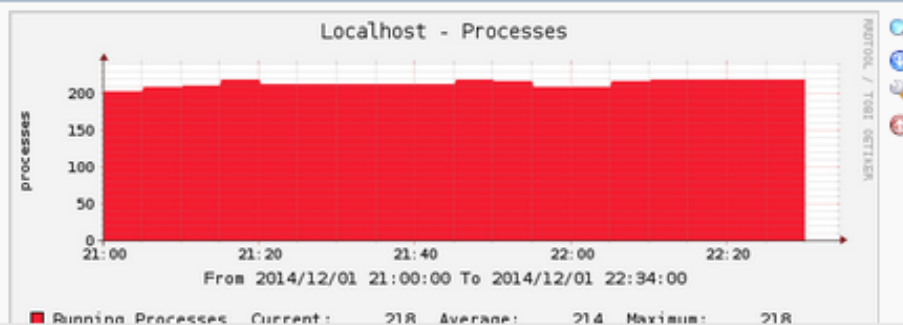
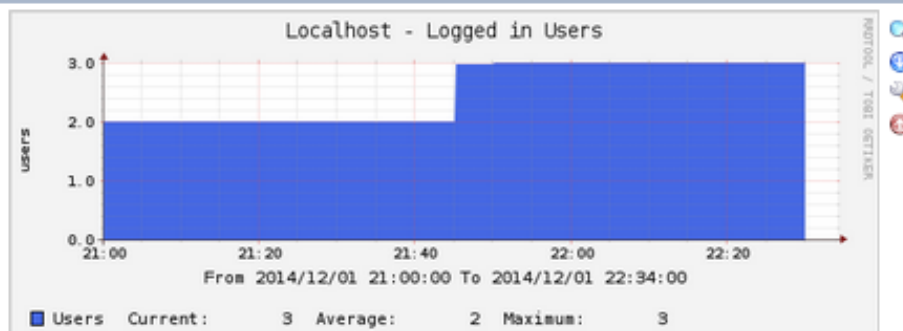
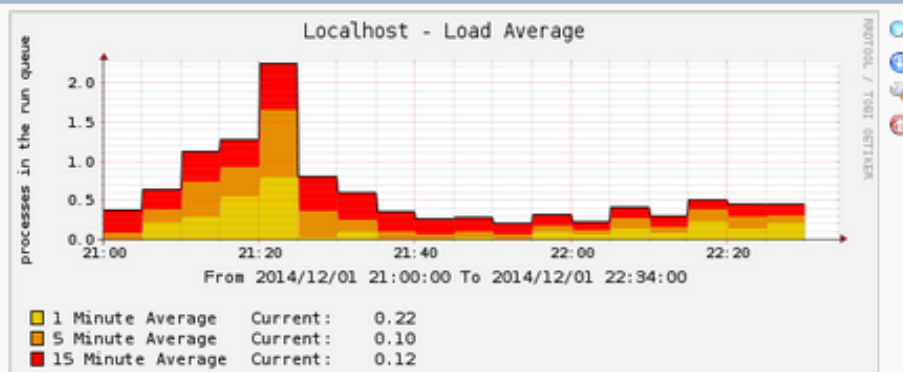
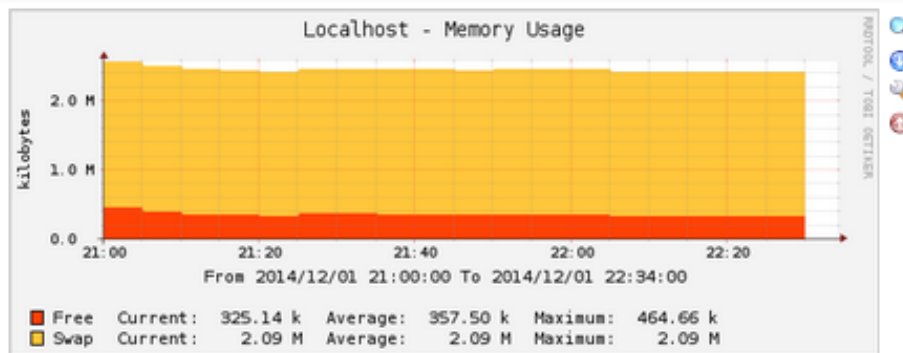


Figura 12.6: Home de cacti.

Podemos configurar parámetros de rango de tiempo o que es lo que queremos muestrear en gráficas, en nuestro caso el uso de memoria, la carga media del servidor, usuario logueados y procesos activos:



+Alt.

Figura 12.7: Gráficas de muestreo en cacti.

13. Desarrolle una página en C o C++ y analice su comportamiento usando valgrind.

Mi código:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <time.h>
5
6  int main(int argc, char **argv){
7
8  if (argc>6 || argc<5){
9  printf("\nERROR. Numero de parametros incorrecto. USO: %s <fila_A> <
    columna_A> <fila_B> <columna_B> [-v].\n", argv[0]);
10 exit(EXIT_FAILURE);
11 }
12
13 int f1 = atoi(argv[1]);
14 int c1 = atoi(argv[2]);
15 int f2 = atoi(argv[3]);
16 int c2 = atoi(argv[4]);
17
18 if(c1!=f2){
19 printf("\nERROR. Las matrices no son multiplicables. Tamano
    incorrecto.\n");
20 exit(EXIT_FAILURE);
21 }
22
23 int i, j, k;
24 int suma;
25 srand(time(NULL));
26
27 //Se crean las matrices y se reserva la memoria
28 int **A;
29 int **B;
30 int **C;
31
32 double tiempo;
33 struct timespec tiempoI, tiempoF;
34
35 A = (int **)malloc(f1*sizeof(int *));
36 for(i=0;i<f1;i++)
37 A[i]=(int *)malloc(c2*sizeof(int));
38
39 B = (int **)malloc(f1*sizeof(int *));
40 for(i=0;i<f1;i++)
41 B[i]=(int *)malloc(c1*sizeof(int));
```

```

42
43 C = (int **)malloc(f2*sizeof(int *));
44 for(i=0;i<f2;i++)
45 C[i]=(int *)malloc(c2*sizeof(int));
46
47
48 //Se rellenan con enteros aleatorios del 0 al 9
49 for (i=0; i<f1; i++)
50 for (j=0; j<c1; j++)
51 B[i][j] = rand() %10;
52
53 for (i=0; i<f2; i++)
54 for (j=0; j<c2; j++)
55 C[i][j] = rand() %10;
56
57
58 //Se multiplica
59 clock_gettime(CLOCK_REALTIME, &tiempoI);
60 for (i=0; i<f1; i++)
61 for (j=0; j<c2; j++){
62 suma = 0;
63 for (k=0; k<c1; k++)
64 suma += B[i][k] * C[k][j];
65 A[i][j] = suma;
66 }
67 clock_gettime(CLOCK_REALTIME, &tiempoF);
68 //Se muestra por pantalla la Solucion Completa
69 if (argc==6){
70 if (strcmp("-v",argv[5])==0){
71 for (i=0; i<f1; i++){
72 printf("\n");
73 for (j=0; j<c2; j++){
74 if (A[i][j]>=1000)
75 printf("%d",A[i][j]);
76 else if (A[i][j]>=100)
77 printf(" %d",A[i][j]);
78 else if (A[i][j]>=10)
79 printf("  %d",A[i][j]);
80 else
81 printf("   %d",A[i][j]);
82 }
83 }
84 printf("\n");
85 }
86 else{
87 printf("Parametro opcional incorrecto.\n");
88 }
89 }
90 else{

```

```

91 tiempo = (double) (tiempoF.tv_sec - tiempoI.tv_sec) + (double) ((
    tiempoF.tv_nsec - tiempoI.tv_nsec) / (1.e+9));
92
93 printf("Tiempo: %8.6f\n", tiempo);
94 printf("Componente(0,0): %d\n", A[0][0]);
95 printf("Componente(N-1,N-1): %d\n", A[f1-1][c2-1]);
96 }
97
98 //Se libera la memoria
99
100 for(i=0; i<f1; i++)
101     free(A[i]);
102 free(A);
103
104 for(i=0; i<f1; i++)
105     free(B[i]);
106 free(B);
107
108 for(i=0; i<f2; i++)
109     free(C[i]);
110 free(C);
111
112 return(EXIT_SUCCESS);
113 }

```

y compilamos con:

```
$ gcc -o matrices matrices.c
```

Y finalmente ejecutamos para ver el resultado de valgrind con los parametros adecuados:

```
root@Server32: ~  
File Edit View Search Terminal Help  
root@Server32:~# valgrind --leak-check=yes ./matrices 10 10 10 10 -v  
==11679== Memcheck, a memory error detector  
==11679== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al.  
==11679== Using Valgrind-3.10.0.SVN and LibVEX; rerun with -h for copyright info  
==11679== Command: ./matrices 10 10 10 10 -v  
==11679==  
278 317 249 214 243 275 260 259 213 247  
304 297 268 209 274 264 194 239 255 237  
125 129 176 92 138 186 81 99 108 101  
249 282 228 178 222 194 154 173 218 216  
232 154 220 179 155 217 162 231 218 106  
253 269 201 162 199 171 177 211 211 213  
185 92 153 164 117 136 131 169 169 70  
123 126 93 81 94 99 67 59 114 106  
182 209 184 165 174 228 163 183 174 135  
235 235 177 253 211 246 268 281 192 161  
==11679==  
==11679== HEAP SUMMARY:  
==11679==      in use at exit: 0 bytes in 0 blocks  
==11679==    total heap usage: 33 allocs, 33 frees, 1,320 bytes allocated  
==11679==  
==11679== All heap blocks were freed -- no leaks are possible  
==11679==  
==11679== For counts of detected and suppressed errors, rerun with: -v  
==11679== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)  
root@Server32:~#
```

Figura 13.1: Resultado en consola de valgring sobre un programa de multiplicación de matrices.

14. Seleccione, instale y ejecute uno, comente los resultados.

Hemos elegido un benchmark llamado "hdparm-read" que prueba la velocidad de lectura de los dispositivos de almacenamiento. Incluso puede probar por dispositivos lógicos y no solo físicos. Primero lo instalamos:

```
zedwarck@Server32:~$ sudo phoronix-test-suite install hdparm-read
Phoronix Test Suite v4.8.3

To Install: pts/hdparm-read-1.0.0

Determining File Requirements .....
Searching Download Caches .....

1 Test To Install

pts/hdparm-read-1.0.0:
  Test Installation 1 of 1
  Installing Test @ 22:12:51

zedwarck@Server32:~$
```

Figura 14.1: Instalación del benchmark hdparm-read

Luego lo ejecutamos:

```
zedwarck@Server32:~$ sudo phoronix-test-suite benchmark hdparm-read
[sudo] password for zedwarck:
no talloc stackframe at ../source3/param/loadparm.c:4864, leaking memory

Phoronix Test Suite v4.8.3

Installed: pts/hdparm-read-1.0.0

hdparm Timed Disk Reads:
pts/hdparm-read-1.0.0
Disk Test Configuration
  1: /dev/sda
  2: /dev/sda1
  3: /dev/sda2
  4: /dev/sda5
  5: Test All Options
Disk To Read: █
```

Figura 14.2: Ejecución del benchmark hdparm-read

Después de elegir el sistema físico-lógico que queremos probar nos pregunta si queremos guar-

dar los resultados del test(También nos muestra un resumen general del sistema).

```
System Information

Hardware:
Processor: Intel Core 2 Quad Q9300 @ 2.50GHz (2 Cores), Motherboard: Intel 440BX
, Chipset: Intel 440BX/ZX/DX, Memory: 1 x 2048 MB DRAM, Disk: 21GB VMware Virtua
l S, Graphics: VMware SVGA II, Audio: Ensoniq ES1371 / Creative, Network: AMD 79
c970

Software:
OS: Ubuntu 14.04, Kernel: 3.13.0-32-generic (i686), Desktop: GNOME Shell 3.10.4,
Display Server: X Server 1.15.1, Display Driver: vmware 13.0.2, OpenGL: 2.1 Mes
sa 10.1.3 Gallium 0.4, Compiler: GCC 4.8, File-System: ext4, Screen Resolution: 6
40x480, System Layer: VMware

Would you like to save these test results (Y/n):
```

Figura 14.3: Resumen de nuestro sistema y confirmación de guardado de datos.

El benchmark realiza cinco pasadas en nuestro caso:

```
40x480, System Layer: VMware

Would you like to save these test results (Y/n): y
Enter a name to save these results under: res-hdparm
Enter a unique name to describe this test run / configuration: test1

If you wish, enter a new description below to better describe this result set /
system configuration under test.
Press ENTER to proceed without changes.

Current Description: VMware testing on Ubuntu 14.04 via the Phoronix Test Suite.
New Description: Test sd1

hdparm Timed Disk Reads:
pts/hdparm-read-1.0.0 [Disk To Read: /dev/sd1]
Test 1 of 1
Estimated Trial Run Count: 5
Estimated Time To Completion: 7 Minutes
Started Run 1 @ 12:24:52
Started Run 2 @ 12:25:02
Started Run 3 @ 12:25:11
Started Run 4 @ 12:25:19
```

Figura 14.4: Realización de los test de disco.

Finalmente nos muestra los resultados (Y nos pregunta si queremos visualizarlos en navegador.):

```
Press ENTER to proceed without changes.
Current Description: VMware testing on Ubuntu 14.04 via the Phoronix Test Suite.
New Description: Test sdal

hdparm Timed Disk Reads:
pts/hdparm-read-1.0.0 [Disk To Read: /dev/sdal]
Test 1 of 1
Estimated Trial Run Count: 5
Estimated Time To Completion: 7 Minutes
Started Run 1 @ 12:24:52
Started Run 2 @ 12:25:02
Started Run 3 @ 12:25:11
Started Run 4 @ 12:25:19
Started Run 5 @ 12:25:28 [Std. Dev: 37.50%]
Started Run 6 @ 12:25:36 [Std. Dev: 33.67%]
Started Run 7 @ 12:25:45 [Std. Dev: 30.54%]
Started Run 8 @ 12:25:53 [Std. Dev: 28.18%]
Started Run 9 @ 12:26:02 [Std. Dev: 26.44%]
Started Run 10 @ 12:26:10 [Std. Dev: 24.93%]

Test Results:
19.08
53.04
64.39
64.85
67.4
67.3
62.24
64.16
67.36
58.99

Average: 58.88 MB/s

Do you want to view the results in your web browser (Y/n):
```

Figura 14.5: Resultado del benchmark hdparm-read.

Resultados vistos en gráfica desde navegador.

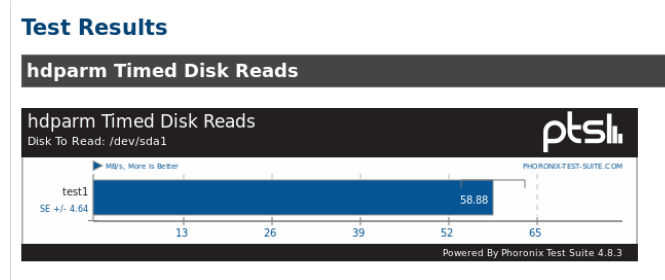


Figura 14.6: Resultado del benchmark hdparm-read en gráfica.

15. ¿Qué es Scala? Instale Gatling y pruebe los escenarios por defecto. [6]

Es un lenguaje de programación orientado a objetos. Se ejecuta en la máquina virtual de Java, por lo que puede integrarse con facilidad en los proyectos Java.

Para instalarlo en Ubuntu basta con: `sudo apt-get install gatling`

16. Lea el artículo sobre Jmeter y elabore un breve resumen.

El documento pretende comparar JMeter con Gatling, en la comparación tomas diversas medidas en un mismo caso simulado con 10000 usuarios y 30000 peticiones por minuto sobre un servidor web nginx.

Después de tomar las medidas hacen una valoración y se llega a la conclusión de que ambos son muy parecidos respecto al rendimiento. Solo destacar el nivel de procesamiento de JMeter es mayor debido a que se ejecuta sobre una maquina virtual java y eso produce un mayor volumen de procesamiento y carga de memoria RAM del servidor.

17. Seleccione un benchmark entre SisoftSandra y Aida. Ejecútelo y muestre capturas de pantalla comentando los resultados.

Después de instalarlo lo ejecutamos y tenemos:

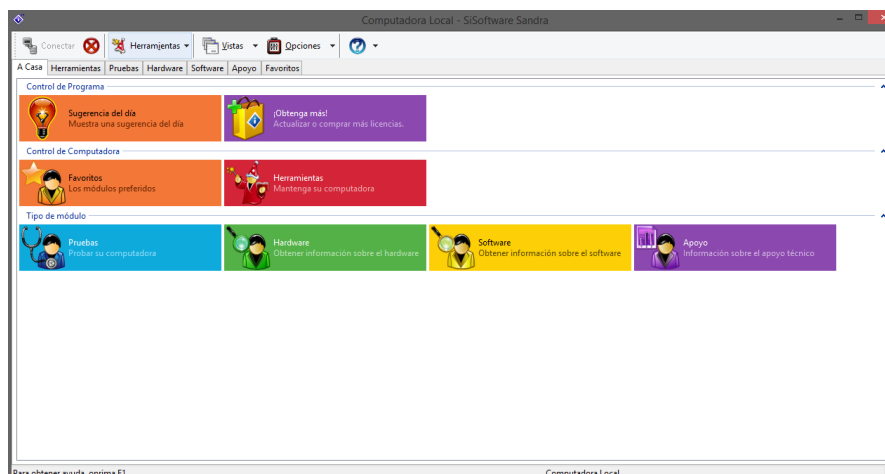


Figura 17.1: Pantalla inicial de SisoftSandra.

Al darle a pruebas para testear nos muestra muchos tipos de tests que puede hacer a nuestra maquina. En nuestro caso elegimos que nos haga un benchmarking de la CPU y sus Gigaflops:

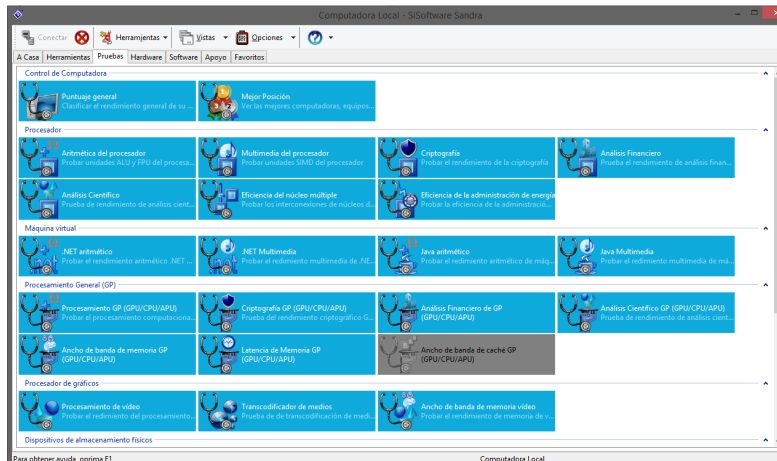


Figura 17.2: Diversos test de SiSoftSandra.

Después de un periodo de tiempo y una vez completado el test nos muestra el resumen y una grafica en la que podemos comparar con otros procesadores del mercado pudiendo elegir nosotros cuales son los procesadores de referencia:

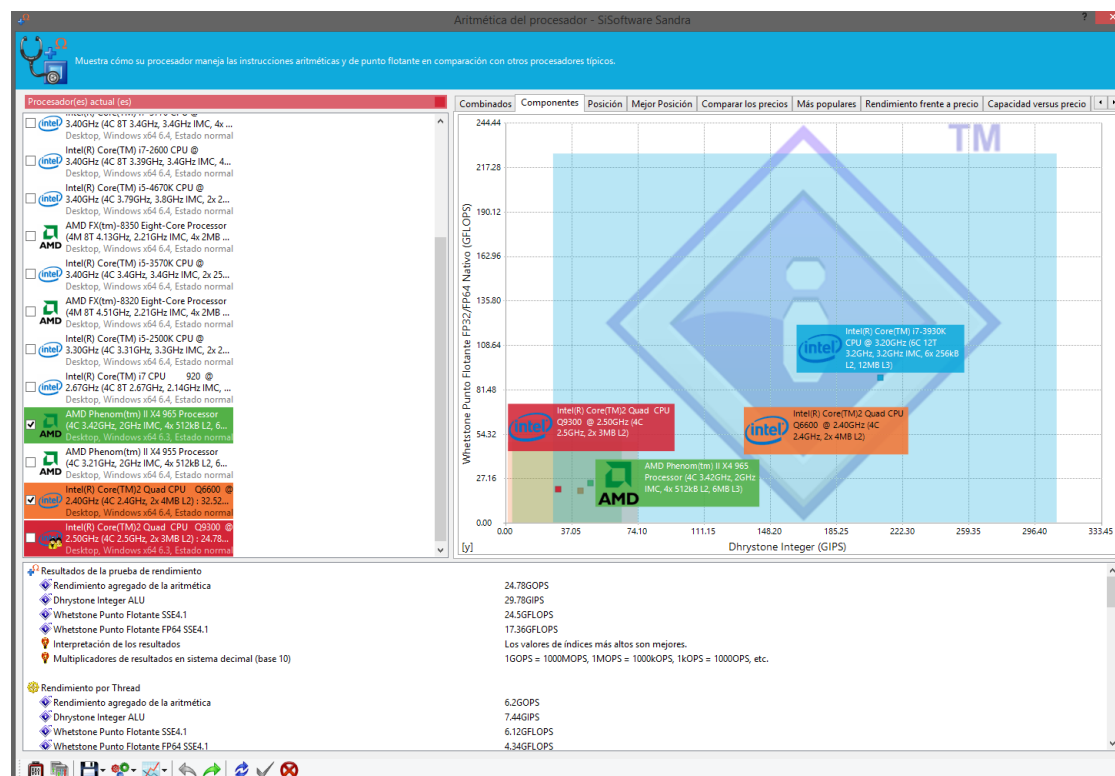


Figura 17.3: Resultado de testear la CPU con SiSoftSandra.

Referencias

- [1] HowToForge. http://www.howtoforge.com/replacing_hard_disks_in_a_raid1_array.
- [2] HowToForge. <http://www.howtoforge.com/how-to-install-tomcat-on-centos-7>.
- [3] MongoDB. <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-red-hat-centos-or-fedora-linux>.
- [4] MongoDB. <http://docs.mongodb.org/manual/tutorial/getting-started>.
- [5] Nagios. http://assets.nagios.com/downloads/nagioscore/docs/Installing_Nagios_Core_From_Source.pdf.
- [6] Scala. <http://docs.scala-lang.org/es/tutorials/scala-for-java-programmers.html>.
- [7] Tecmint. <http://www.tecmint.com/how-to-enable-epel-repository-for-rhel-centos-6-5>.
- [8] thedutchlab.com. <https://thedutchlab.com/en/news/installing-zabbix-on-ubuntu-14-04>.
- [9] Ubuntu. <https://help.ubuntu.com/community/Fail2ban>.
- [10] Ejecutivo Soporte XpressHosting. <http://blog.xpress.com.mx/2011/12/que-es-el-comando-screen-y-como-utilizarlo>.