

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



Práctica 5

José Carlos Martínez Velázquez

28 de diciembre de 2015

Índice

1. Al modificar los valores del kernel de este modo, no logramos que persistan después de reiniciar la máquina. ¿Qué archivo hay que editar para que los cambios sean permanentes? 3
2. ¿Con qué opción se muestran todos los parámetros modificables en tiempo de ejecución? Elija dos parámetros y explique, en dos líneas, qué función tienen. 3
3. Realice una copia de seguridad del registro y restaurela, ilustre el proceso con capturas. 4
4. ¿Cómo se abre una consola en Windows? ¿Qué comando hay que ejecutar para editar el registro? Muestre su ejecución con capturas de pantalla. 6
5. Las cadenas de caracteres y valores numéricos tienen distintos tipos. Busque en la documentación de Microsoft y liste todos los tipos de valores. 7
6. Enumere qué elementos se pueden configurar en Apache y en IIS para que Moodle funcione mejor. 8
7. Ajuste la compresión en el servidor y analice su comportamiento usando varios valores para el tamaño a de archivo partir del cual comprimir. Para comprobar que está comprimiendo puede usar el navegador o comandos como curl (see url) o lynx. Muestre capturas de pantalla de todo el proceso. 9
8. Usted parte de un SO con ciertos parámetros definidos en la instalación (Práctica 1), ya sabe instalar servicios (Práctica 2) y cómo monitorizarlos (Práctica 3) cuando los somete a cargas (Práctica 4). Al igual que ha visto cómo se puede mejorar un servidor web (Práctica 5 Sección 3.1), elija un servicio (el que usted quiera) y modifique un parámetro para mejorar su comportamiento. (9.b) Monitoree el servicio antes y después de la modificación del parámetro aplicando cargas al sistema (antes y después) mostrando los resultados de la monitorización. 11
9. Cuestión opcional 1: Realice lo mismo que en la cuestión 8 pero para otro servicio. 14

1. **Al modificar los valores del kernel de este modo, no logramos que persistan después de reiniciar la máquina. ¿Qué archivo hay que editar para que los cambios sean permanentes?**

Cuando modificamos los valores del kernel a través de `sysctl`, si queremos que los cambios persistan después de reiniciar la máquina, hay que añadir los cambios al archivo `/etc/sysctl.conf`.

Un ejemplo muy sencillo de cómo hacerlo sería cambiar el nombre del host. Para que los cambios sean permanentes, se ejecuta el siguiente comando:

```
:~$ echo "kernel.hostname=miequipo" >> /etc/sysctl.conf && sysctl -p
```

(Fuentes: [3])

2. **¿Con qué opción se muestran todos los parámetros modificables en tiempo de ejecución? Elija dos parámetros y explique, en dos líneas, qué función tienen.**

Los parámetros modificables en tiempo de ejecución se pueden listar ejecutando el comando:

```
:~$ sudo sysctl -a
```

Si ejecutamos el comando obtenemos dicha lista:

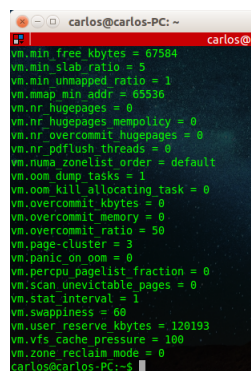


Figura 2.1: Parte final de la lista de parámetros modificables en tiempo de ejecución.

Algunos parámetros que me resultan interesantes son:

- **kernel.panic:** Ante un error interno del que el equipo no se puede recuperar, ¿cuántos segundos deben pasar antes de reiniciar? Para ello ejecutamos

```
:~$ echo "kernel.panic=[segundos_antes_de_reiniciar]" >> /etc/sysctl.conf
      && sysctl -p
```

Lo que permitirá que tras un kernel panic se reinicie al pasar 5 segundos. Por defecto está a 0. Kernel panic a 0 indica que detectar un kernel panic implica un reinicio inmediato.

- **kernel.pid_max**: Sirve para incrementar o disminuir el límite de identificadores de proceso. Para realizar este cambio se debe ejecutar el comando:

```
:~$ echo "kernel.pid_max=[num_max_de_identificadores_de_proceso]" >> /etc/sysctl.conf
      && sysctl -p
```

Por defecto vale 32768.

(Fuentes: [3] [10] [15] [6])

3. Realice una copia de seguridad del registro y restaurela, ilustre el proceso con capturas.

Para realizar copias de seguridad en Windows Server 2012 Server r2, abriremos el editor de registro, para ello pulsamos la tecla Windows + r (ejecutar), escribimos regedit y posteriormente aceptar.

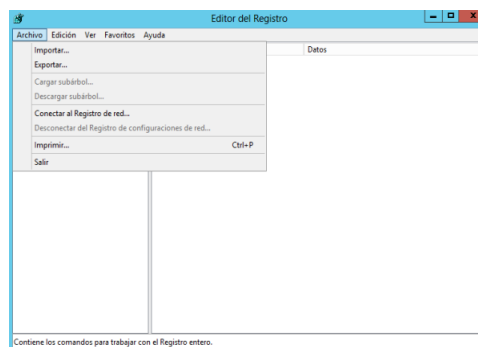


Figura 3.1: Realizando una copia de seguridad del registro.

Ahora en el menú Archivo, elegimos Exportar, que nos permitirá realizar una copia del registro en una determinada ubicación, en este caso, Escritorio.

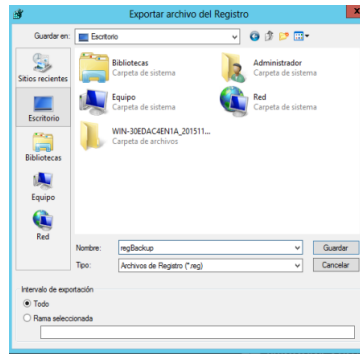
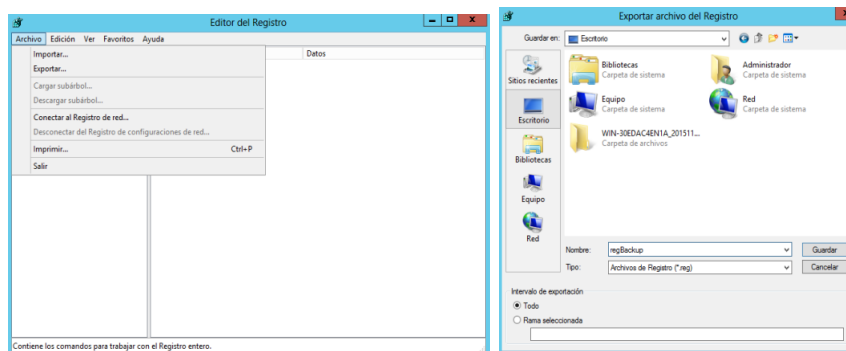


Figura 3.2: Guardando el archivo de copia de seguridad.

Ya tenemos la copia hecha. Supongamos que en este momento tocamos algo y resulta ser una mala configuración. Si no hubiéramos hecho esta copia, podría ser fatal para el servidor y para todos los servicios que prestamos. Pero con la copia hecha, podemos volver atrás. Restauramos la copia. Para ello, volvemos al editor de registro, abrimos el menú Archivo → Importar. Elegimos la copia de seguridad que hicimos y pulsamos aceptar. También se puede hacer con doble clic sobre la copia creada.



(a) Seleccionando copia de seguridad para restaurar. (b) Restaurando el registro.

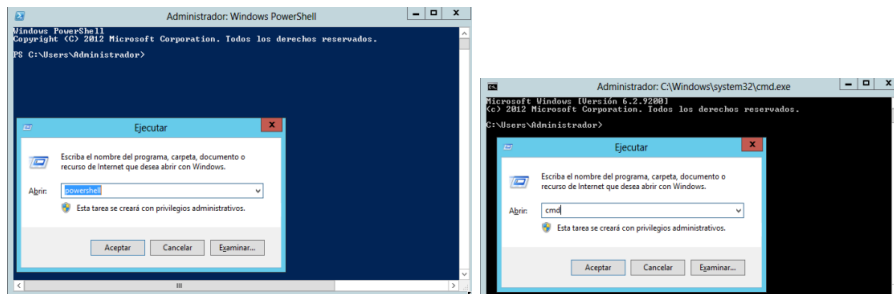
Figura 3.3: Proceso de restauración del registro.

Una vez finalizado el proceso, todo el registro habrá sido restaurado. En casos reales, cuando se realiza una copia de seguridad del registro, conviene asignarle un identificador al nombre, recomendable la fecha en que se hizo.

(Fuentes: [9])

4. ¿Cómo se abre una consola en Windows? ¿Qué comando hay que ejecutar para editar el registro? Muestre su ejecución con capturas de pantalla.

Para abrir una consola (símbolo del sistema) hay que ejecutar cmd. Para ello, pulsamos la combinación de teclas Windows + r (ejecutar) y escribimos cmd. Otro tipo de consola es PowerShell (Windows + r → powershell). Se puede ejecutar indistintamente una u otra.



(a) Ejecución de PowerShell.

(b) Ejecución de cmd.

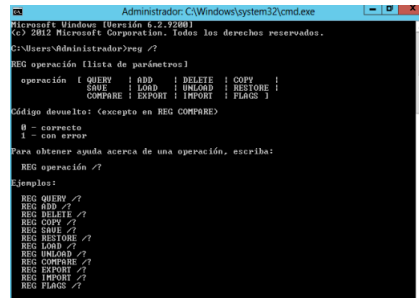
Figura 4.1: Proceso de restauración del registro.

El comando que nos permite editar el registro es reg (disponible en PowerShell y en cmd). Los subcomandos de reg son:

- **REG QUERY**: Busca y devuelve el valor de una clave.
- **REG ADD**: Agrega elementos al registro.
- **REG DELETE**: Elimina elementos del registro.
- **REG COPY**: Copia elementos del registro entre equipos.
- **REG SAVE**: Guarda elementos del registro en archivos con formato .HIV.
- **REG RESTORE**: Restaura elementos del registro.
- **REG LOAD, REG UNLOAD**: Se cargan y transfieren elementos del registro.
- **REG COMPARE**: Compara elementos del registro.
- **REG EXPORT**: Exporta el contenido del registro (copia de seguridad).
- **REG IMPORT**: Importa el contenido del registro (restaura el registro de un archivo .REG).

La secuencia completa de cómo usar reg se puede consultar con el comando:

```
> reg /?
```



```
Administrador: C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.2.9200]
(c) 2012 Microsoft Corporation. Todos los derechos reservados.
C:\Users\Administrador>reg /?

REG operación [lista de parámetros]

operación { QUERY | ADD | DELETE | COPY |
           MOVE | LOAD | UNLOAD | RESTORE |
           COMPARE | EXPORT | IMPORT | PLACES }

Código devuelto: (excepto en REG COMPARE)
0 - correcto
1 - con error

Para obtener ayuda acerca de una operación, escriba:
REG operación /?

Ejemplos:
REG QUERY /?
REG ADD /?
REG DELETE /?
REG COPY /?
REG MOVE /?
REG LOAD /?
REG UNLOAD /?
REG RESTORE /?
REG COMPARE /?
REG EXPORT /?
REG IMPORT /?
REG PLACES /?
```

Figura 4.2: Cómo usar el comando reg.

(Fuentes: [8])

5. Las cadenas de caracteres y valores numéricos tienen distintos tipos. Busque en la documentación de Microsoft y liste todos los tipos de valores.

Los distintos tipos que podemos encontrar, según la documentación de Microsoft son:

- **REG_BINARY** (valor binario): Datos binarios sin formato. La mayoría de la información sobre componentes de hardware se almacena en este tipo de dato.
- **REG_DWORD** (valor DWORD): Son representados por un número de 4 bytes de longitud. Se usan por ejemplo, para representar parámetros de controladores de dispositivo.
- **REG_EXPAND_SZ** (valor alfanumérico expandible): Cadena de datos de longitud variable.
- **REG_MULTI_SZ** (valor de cadena múltiple): Es la más sencilla de leer, pues las entradas aparecen separadas por espacios, comas u otros signos de puntuación.
- **REG_SZ** (valor de cadena): Cadena de texto de longitud fija.
- **REG_RESOURCE_LIST** (valor binario): Conjunto de matrices anidadas diseñada para almacenar una lista de recursos utilizados por el controlador de un dispositivo de hardware o uno de los dispositivos físicos que controla.
- **REG_RESOURCE_REQUIREMENTS_LIST** (valor binario): Serie de matrices anidadas diseñada para almacenar una lista de controladores de dispositivo de posibles recursos de hardware que el controlador pueden utilizar.
- **REG_FULL_RESOURCE_DESCRIPTOR** (valor binario): Serie de matrices anidadas diseñada para almacenar una lista de recursos utilizados por un dispositivo físico de hardware.

- **REG_LINK** (vínculo): Cadena Unicode que da nombre a un vínculo simbólico.
- **REG_QWORD** (valor QWORD): Datos representados por un número entero de 64 bytes.

(Fuentes: [11])

6. Enumere qué elementos se pueden configurar en Apache y en IIS para que Moodle funcione mejor.

La página oficial de Moodle nos hace las siguientes recomendaciones:

Recomendaciones para Apache:

- Se recomienda definir la directiva `MaxClients` en función de la memoria disponible. Un valor orientativo sería dividir el 80 % de la memoria disponible entre la cantidad de memoria máxima que van a utilizar los procesos de Apache.
- Para reducir la memoria consumida, se recomienda cargar el número mínimo de módulos que se permiten, editando el archivo `httpd.conf`
- Bajar el valor de `MaxRequestsPerChild` en `httpd.conf` entre 20 y 30.
- Si el servidor está ejecutando un número considerable de procesos, hay que ajustar la directiva `KeepAlive`, poniéndola a `Off` (sólo si no hay enlaces a recursos). Además, se recomienda bajar el valor de `KeepAliveTimeout` a valores entre 2 y 5, lo que permitirá que una conexión libere recursos si se mantiene inactiva durante ese tiempo. De forma predeterminada son 15 segundos.
- Si no se usa un archivo `.htaccess` (archivo de configuración de directorios) hay que cambiar el valor de la directiva `AllowOverride` a `None` para que se ahorre tiempo, evitando que se busque el archivo `.htaccess`.
- Ajustar la directiva `DirectoryIndex` para evitar que busque cuál es la página principal cuando se solicite el `index`, colocando directamente el recurso que se debe devolver:

<code>DirectoryIndex index.php index.html index.htm</code>
--

- Si no se están realizando trabajos en un servidor de desarrollo, se recomienda cambiar la directiva `ExtendedStatus` a `Off` y deshabilitar `mod_info` y `mod_status`.
- Es necesario reducir la latencia DNS.
- Cambiar el valor de `Timeout` entre 30 y 60 (segundos).
- Para la directiva `Options` hay que evitar `Options Multiviews` ya que escanea el directorio completo. Esto reducirá las peticiones de E/S que se hacen.

En IIS (Internet Information Services) no se pueden cambiar tantas cosas como en apache. Se listan a continuación las mejoras que se pueden hacer.

Recomendaciones para IIS:

Todos los cambios a realizar hay que hacerlos sobre la clave de registro que se encuentra en HKLM\SYSTEM\CurrentControlSet\Services\Inetinfo\Parameters

- Para conseguir el mismo resultado que cambiando KeepAliveTimeout en Apache, Hay que cambiar el valor de ListenBackLog entre 2 y 5.
- Hay que cambiar el valor de MemCacheSize, ajustándolo aproximadamente a la mitad de la memoria disponible.
- Hay que ajustar el valor máximo que puede tener un archivo almacenado en caché (en Bytes). Para ello se cambia el valor de MaxCachedFileSize.
- Por último, hay que crear un valor DWORD (cuestión 5), llamado ObjectCacheTTL que definirá cuánto tiempo (en milisegundos) se guardarán los objetos de la caché en la memoria. De forma predeterminada es 30000 (30 segundos).

(Fuentes: [5])

7. Ajuste la compresión en el servidor y analice su comportamiento usando varios valores para el tamaño a de archivo partir del cual comprimir. Para comprobar que está comprimiendo puede usar el navegador o comandos como curl (see url) o lynx. Muestre capturas de pantalla de todo el proceso.

Lo primero que debemos hacer es ir al Administrador de IIS, se puede buscar en el menú de inicio. Hacemos clic y debería aparecer el siguiente panel:

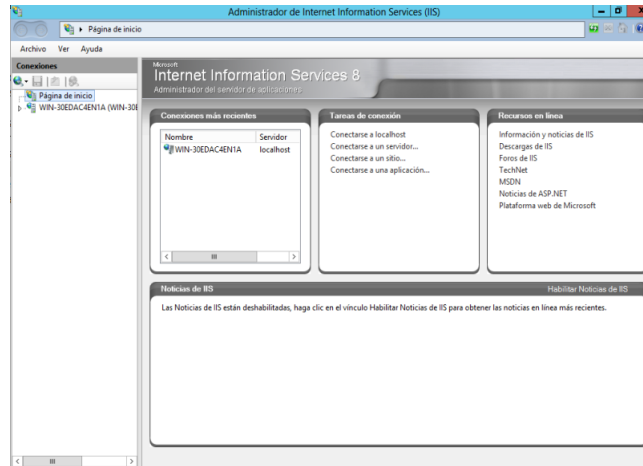
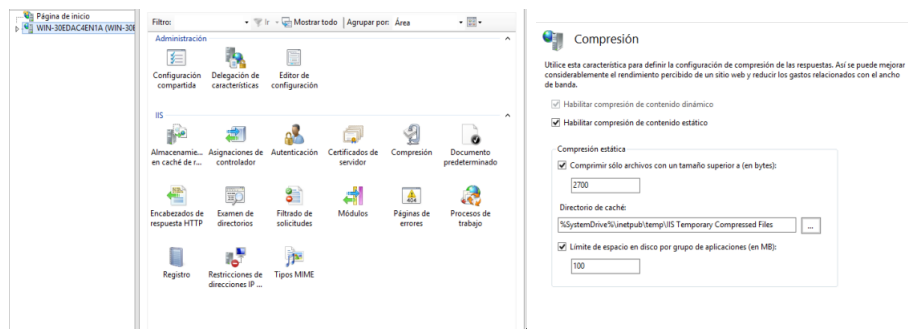


Figura 7.1: Panel de administración de IIS.

En la parte izquierda hacemos clic en el servidor y en el panel que aparece a continuación, hacemos doble clic en compresión:



(a) Opciones del servidor en el panel de administración.

(b) Panel de compresión.

Figura 7.2: Accediendo al panel de compresión.

Como vemos en la figura 7.2, en este caso sólo podemos habilitar la compresión de contenido estático. Tal y como nos recomiendan en [12]:

- En el cuadro Comprimir sólo archivos con un tamaño superior a (en bytes), hay que poner el tamaño de archivo mínimo que IIS debe comprimir. Para este ejemplo, lo dejamos en 256B.
- En el cuadro de texto Directorio de caché hay que escribir la ruta de un directorio temporal en el que una vez comprimido un archivo estático, se almacenará en la caché de éste directorio hasta que expire o cambie su contenido. Lo dejamos por defecto.

- c) Por último, opcionalmente se puede definir un límite de espacio en disco para cada grupo de aplicaciones. Lo dejamos por defecto.

Ahora en la parte derecha, clicamos en Aplicar. La compresión ha sido habilitada. Vamos a probar la configuración. Abrimos el navegador desde la máquina anfitriona y monitorizamos las peticiones. Vemos como para una petición, en su cabecera, el apartado accept-encoding nos indica que esta siendo comprimido con gzip:

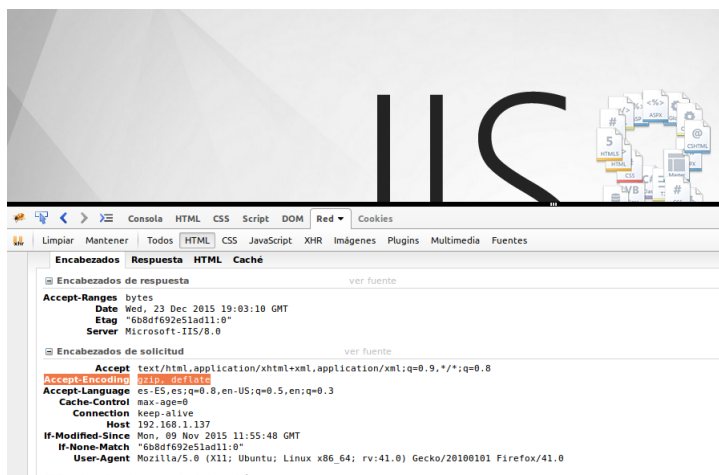


Figura 7.3: Comprobando la compresión.

(Fuentes: [12])

8. Usted parte de un SO con ciertos parámetros definidos en la instalación (Práctica 1), ya sabe instalar servicios (Práctica 2) y cómo monitorizarlos (Práctica 3) cuando los somete a cargas (Práctica 4). Al igual que ha visto cómo se puede mejorar un servidor web (Práctica 5 Sección 3.1), elija un servicio (el que usted quiera) y modifique un parámetro para mejorar su comportamiento. (9.b) Monitoree el servicio antes y después de la modificación del parámetro aplicando cargas al sistema (antes y después) mostrando los resultados de la monitorización.

En esta cuestión vamos a configurar el rendimiento del servidor Apache instalado en la máquina Ubuntu Server que hemos usado durante el desarrollo de las prácticas de

Ingeniería de Servidores. Esta máquina tiene instalado un LAMP, que contiene Apache, por lo que podemos hacer peticiones HTTP con Apache Benchmark (ab) y medir su rendimiento.

Lo primero que debemos comprobar es que el servidor y la máquina que va a lanzar Apache Benchmark estén en la misma subred. Vamos a crear en Virtual Box un adaptador sólo-anfitrión, de manera que todas las máquinas queden conectadas entre sí y asegurarnos el acceso. Un adaptador sólo-anfitrión es una subred para conectar máquinas, de éste modo nos ahorraremos asignar IPs estáticas y nos evitaremos complicaciones si una red tiene particularidades o mecanismos de seguridad que impidan este tipo de comunicaciones o no tenemos acceso a internet. En cada red habrá que tomar medidas distintas y no es el objetivo de esta cuestión estudiarlas.

```

(a) IP para la interfaz vboxnet0 en la máquina Ubuntu Server.
(b) IP para la interfaz vboxnet0 en la máquina Anfitriona (Ubuntu 14.04).

```

Figura 8.1: Configuración de red de servidor y cliente.

Como se puede observar, las máquinas están en la subred 192.168.56.xx, por lo que será sencillo comunicarse.

Desde la máquina anfitriona, vamos a lanzar Apache Benchmark con un numero de peticiones concurrentes de 10 a la vez y un número total de peticiones de 1000. Ejecutaremos, pues, el siguiente comando:

```

:~$ab -c 10 -n 1000 http://192.168.56.101/

```

Los resultados antes de cambiar parámetros son los siguientes:

```

(a) Resultados Apache Benchmark (I).
(b) Resultados Apache Benchmark (II).

```

Figura 8.2: Rendimiento del servidor antes de cambiar parámetros.

Como podemos observar en la figura 8.2, se han transferido 11,237 MB (11783000 B) en 0,669 segundos, con lo que se obtiene una tasa de $\frac{11,237 \text{ MB}}{0,669 \text{ seg}} = 16,797 \text{ MB/seg}$, lo que se puede considerar ya de por sí una buena tasa. Nuestro objetivo es mejorarla. Para ello vamos a adoptar la configuración que se nos recomienda en [1], donde editando el archivo `/etc/apache2/apache2.conf`, hay que añadir las siguientes líneas:

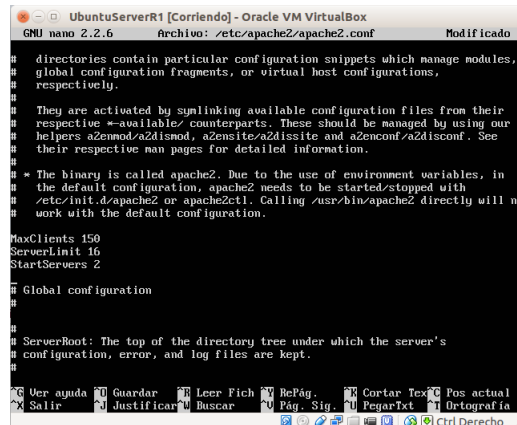


Figura 8.3: Cambiando algunos parámetros de Apache.

Por supuesto, **hay que reiniciar el servicio apache2** para que los cambios surtan efecto con el comando:

```
~$sudo service apache2 restart
```

De nuevo, desde la máquina cliente, vamos a lanzar Apache Benchmark a ver qué hemos conseguido con esta configuración. Los resultados, tras realizar los cambios, son los siguientes:

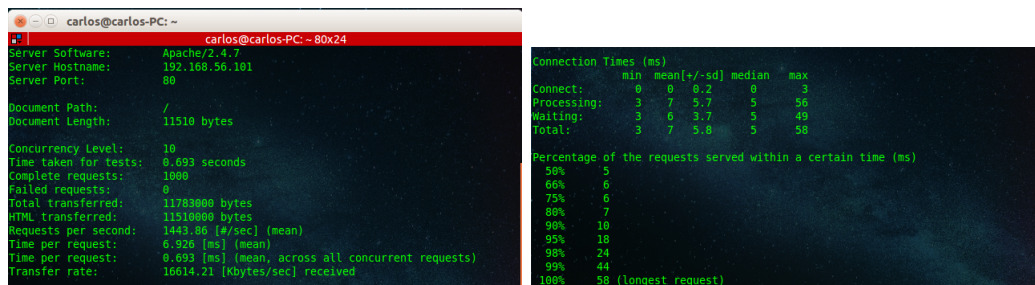


Figura 8.4: Rendimiento del servidor después de cambiar parámetros.

Como podemos observar en la figura 8.4, después de haber cambiado la configuración a la recomendada por [1], se han transferido 11,237 MB (11783000 B) en 0,708 segundos, con lo que se obtiene una tasa de $\frac{11,237 \text{ MB}}{0,693 \text{ seg}} = 16,215 \text{ MB/seg}$. Podemos llegar a la

conclusión de que los cambios no han sido muy efectivos. Hemos empeorado un poco, pero ésto se puede deber a razones aleatorias del estado del servidor, habría que estudiarlo de forma estadística. En cualquier caso, la ganancia o la pérdida en rendimiento con estos nuevos cambios no parece significativa pero me quedaría con la configuración anterior.

(Fuentes: [1] [2])

9. Cuestión opcional 1: Realice lo mismo que en la cuestión 8 pero para otro servicio.

Vamos a usar y ampliar lo que hemos realizado en la cuestión 8. El objetivo en esta cuestión es crear un servidor de nombres de dominio (DNS). Voy a utilizar 4 máquinas: la anfitriona y las tres virtuales usadas durante el desarrollo de la totalidad de las prácticas. Usaré también el adaptador sólo-anfitrión creado en la cuestión 8.

Lo primero que habrá que comprobar es que todas las máquinas están en la misma subred.

(a) IP para la interfaz vboxnet0 en la máquina Ubuntu Server. (b) IP para la interfaz vboxnet0 en la máquina Windows Server 2012 r2.

(c) IP para la interfaz vboxnet0 en la máquina CentOS. (d) IP para la interfaz vboxnet0 en la máquina Anfitriona (Ubuntu 14.04).

Figura 9.1: Configuración de red de todas las máquinas.

Al igual que en la cuestión 8, hemos comprobado que las máquinas están en la subred 192.168.56.xx y pueden comunicarse. Una vez comprobado (que es importante para des-

cartar que ha ocurrido un error por culpa de ésto), vamos a proceder con la instalación y configuración del servidor DNS.

Por preferencia propia, he decidido que el servidor DNS sea la máquina virtual Ubuntu Server. Para instalar y configurar un servidor DNS en una máquina ubuntu es necesario el paquete bind9. Instalaremos el paquete con el comando:

```
:~$sudo apt-get install bind9
```

Antes de instalarlo, quizás conviene comprobar que no está instalado, para lo que se puede ejecutar el comando:

```
:~$dpkg -L bind9
```

(Nota: si se ha configurado el adaptador sólo-anfitrión y sólo se tiene dicha interfaz, entonces debemos configurar de nuevo la red de la máquina virtual para poder acceder a los repositorios de Ubuntu en internet y descargar el paquete bind9, ya sea con conexión NAT o Bridged, luego volver a habilitar el adaptador sólo-anfitrión.)

Una vez instalado bind9, vamos a editar algunos archivos de Bind, que controlan la configuración del servicio de nombres de dominio. Personalmente **usaré ssh -X a través de mi máquina anfitriona** para poder usar la interfaz gráfica de gedit, porque no me gusta vim.

El primer archivo que editaremos es /etc/bind/named.conf.options en el que descomentaremos el bloque forwarders, indicando la IP del servidor de nombres donde consultará nuestro servirdor si no conoce la respuesta. Usaremos la dirección DNS pública de Google: 8.8.8.8.

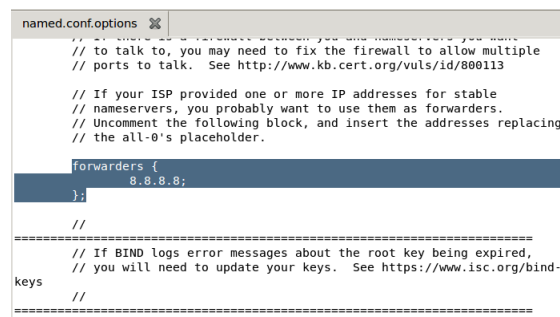
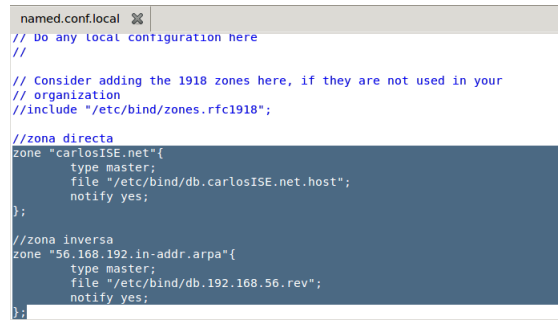


Figura 9.2: Edición de /etc/bind/named.conf.options.

El siguiente paso es editar el archivo /etc/bind/named.conf.local, donde vamos a especificar las zonas de búsqueda directa e inversa de nuestro servicio DNS. La búsqueda directa consiste en traducir nombres de dominio en direcciones IP. La búsqueda inversa consiste en traducir direcciones IP en nombres de dominio. También aquí vamos a definir el tipo del servicio: maestro o esclavo.



```
named.conf.local
// Do any local configuration here
//
// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

//zona directa
zone "carlosISE.net"{
    type master;
    file "/etc/bind/db.carlosISE.net.host";
    notify yes;
};

//zona inversa
zone "56.168.192.in-addr.arpa"{
    type master;
    file "/etc/bind/db.192.168.56.rev";
    notify yes;
};
```

Figura 9.3: Edición de `/etc/bind/named.conf.local`.

Aquí debemos pararnos un poco. Los dos bloques que se aprecian seleccionados en la figura 9.3 hay que añadirlos "a mano". El bloque de la zona directa es sencillo de comprender. Se declara un bloque zona con la palabra "zone", el nombre del dominio va entre comillas y a continuación definimos que es de tipo maestro con "type". La línea file indica qué archivo contiene las tablas de traducción directa (habrá que crearlo). Con la línea notify yes, cuando una zona se carga o cambia se envían los mensajes de notificación a los servidores definidos en el archivo de la zona, en éste caso, `/etc/bind/db.carlosISE.net.host`.

El bloque de zona inversa se declara igual, pero tiene una particularidad. El nombre de la zona debe contener la dirección de subred al revés, seguida de *in-addr.arpa*, que es el dominio estándar reservado para DNS por IPv4.

Para comprobar que no tenemos errores deberíamos ejecutar el comando:

```
:~$named-checkconf /etc/bind/named.conf.local
```

Si no devuelve nada, todo está correcto. Si devuelve algo, será sencillo encontrar el error. Por lo general, se nos olvida el carácter ";" después de cada sentencia.

Ya que todo está correcto, vamos a crear los archivos que contienen las tablas de búsqueda. Deben tener los nombres que hemos definido en `/etc/bind/named.conf.local`. En este caso, el archivo que contiene las tablas de búsqueda directa va a ser `/etc/bind/db.carlosISE.net.host` y el de búsqueda inversa será `/etc/bind/db.192.168.56.rev`. Ejecutamos los comandos:

```
:~$sudo touch /etc/bind/db.carlosISE.net.host
:~$sudo touch /etc/bind/db.192.168.56.rev
```

Ahora los editaremos como sigue. El archivo `/etc/bind/db.carlosISE.net.host` deberá contener lo siguiente:


```

db.carlosISE.net.host
;
; BIND data file for local loopback interface
;
$TTL 604800
@      IN      SOA      carlosISE.net.  root.carlosISE.net. (
                                1          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800 )   ; Negative Cache TTL
;
@      IN      NS       carlosISE.net.
carlosISE.net.      IN      A       192.168.56.101
wserver2012r2.carlosISE.net.  IN      A       192.168.56.140
centos.carlosISE.net      IN      A       192.168.56.102
anfitriona.carlosISE.net.   IN      A       192.168.56.1

```

Figura 9.4: Edición de /etc/bind/db.carlosISE.net.host.

El archivo /etc/bind/db.192.168.56.rev debe contener lo siguiente:

```

db.192.168.56.rev
;
; BIND data file for local loopback interface
;
$TTL 604800
@      IN      SOA      carlosISE.net.  root.carlosISE.net. (
                                1          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800 )   ; Negative Cache TTL
;
@      IN      NS       carlosISE.net.
101    IN      PTR      carlosISE.net.
140    IN      PTR      wserver2012r2.carlosISE.net.
102    IN      PTR      centos.carlosISE.net
1      IN      PTR      anfitriona.carlosISE.net.

```

Figura 9.5: Edición de /etc/bind/db.192.168.56.rev.

Donde la primera columna es el ID de dispositivo de la dirección IP de cada máquina (el último octeto de cada dirección IP. Véase figura 9.1).

Una vez hecho, para comprobar que no hay errores, deberíamos ejecutar los comandos:

```

:~$sudo named-checkzone /etc/bind/db.carlosISE.net.host
:~$sudo named-checkzone /etc/bind/db.192.168.56.rev

```

Debiendo obtener los siguientes resultados:

```

carlos@ubuntu: /etc/bind
carlos@ubuntu: /etc/bind 80x24
carlos@ubuntu:/etc/bind$ named-checkzone carlosISE.net db.carlosISE.net.host
zone carlosISE.net/IN: loaded serial 1
OK
carlos@ubuntu:/etc/bind$ named-checkzone carlosISE db.192.168.56.rev
zone carlosISE/IN: loaded serial 1
OK
carlos@ubuntu:/etc/bind$

```

Figura 9.6: Comprobando errores en las tablas de DNS.

El siguiente archivo que editaremos será /etc/resolv.conf, donde vamos a definir la dirección IP del servidor y el dominio donde se van a realizar las búsquedas.

```

resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
# DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 192.168.56.101
search carlosISE.net

```

Figura 9.7: Definiendo la IP y el nombre del dominio de DNS.

Ahora **hay que reiniciar el servicio bind9** para que todos los cambios tengan efecto. Ejecutamos el comando:

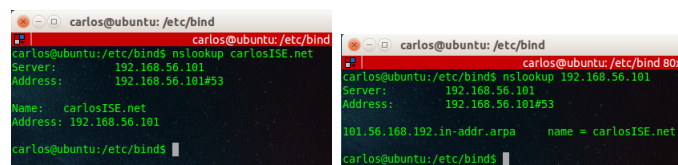
```
:~$sudo service bind9 restart
```

La configuración del servidor DNS ha terminado. Ha llegado la hora de la verdad: probarlo. Lo haremos con el comando nslookup. Probaremos las dos zonas de búsqueda directa e inversa. Para probar la zona directa ejecutaremos el comando:

```
:~$sudo nslookup carlosISE.net
```

Y para probar la zona inversa ejecutamos el comando:

```
:~$sudo nslookup 192.168.56.101
```



```

carlos@ubuntu: /etc/bind
carlos@ubuntu: /etc/bind$ nslookup carlosISE.net
Server:      192.168.56.101
Address:     192.168.56.101#53
Name:   carlosISE.net
Address: 192.168.56.101
carlos@ubuntu: /etc/bind$

carlos@ubuntu: /etc/bind
carlos@ubuntu: /etc/bind$ nslookup 192.168.56.101
Server:      192.168.56.101
Address:     192.168.56.101#53
101.56.168.192.in-addr.arpa    name = carlosISE.net.
carlos@ubuntu: /etc/bind$

```

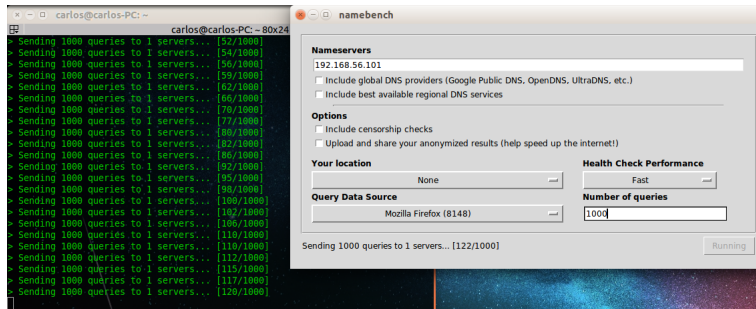
(a) Probando la zona directa. (b) Probando la zona inversa.

Figura 9.8: Prueba de la configuración del servidor DNS.

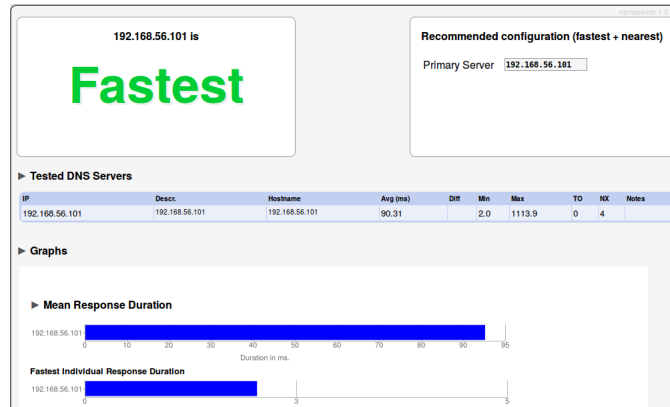
Bien, parece que todo funciona. Ahora vamos a lanzar un benchmark contra el servidor desde la máquina anfitriona. He instalado un benchmark específico para DNS, que se puede instalar con el comando:

```
:~$sudo apt-get install namebench
```

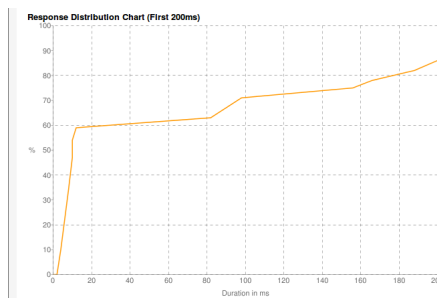
Como sabemos la IP del servidor (192.168.56.101) vamos a lanzar 1000 consultas DNS al mismo (para que la muestra sea significativa) y evaluar sus prestaciones (ejecutando el comando namebench y configurándolo a través de su interfaz gráfica). Los resultados obtenidos son los siguientes:



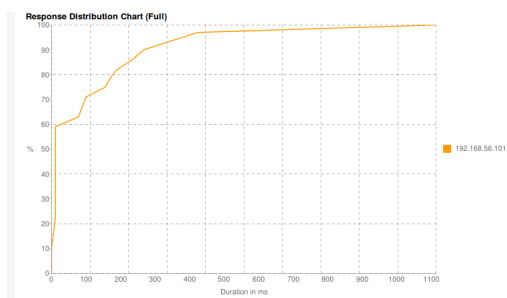
(a) Lanzando peticiones DNS al servidor.



(b) Resultados del benchmark (I).



(c) Resultados del benchmark (II).



(d) Resultados del benchmark (III).

► Configuration

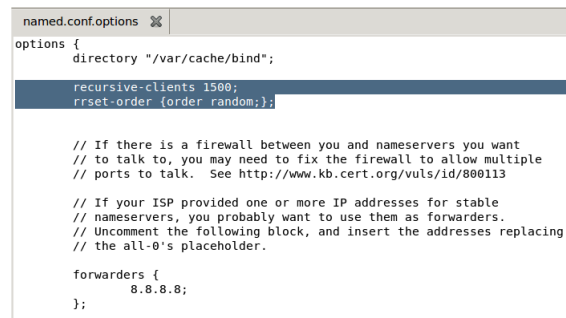
Name	Value
benchmark_thread_count	2
enable_censorship_checks	0
health_thread_count	40
health_timeout	3.75
hide_results	0
input_source	reflex
num_servers	11
ping_timeout	0.5
query_count	1000
run_count	1
select_mode	automatic
template	html
timeout	3.5
upload_results	0
version	1.3.1

(e) Resultados del benchmark (IV).

Figura 9.9: Resultados de lanzar DNS Benchmark contra el servidor DNS.

Nos centraremos en que, de media, el servidor tarda 90.31 ms en resolver direcciones. Nuestro objetivo es bajar ese tiempo cambiando ciertos parámetros. Todos los parámetros que se cambian serán líneas que se añaden al archivo `/etc/bind/named.conf.options`.

Lo primero que haremos será aumentar el número de clientes simultáneos con la línea `recursive-clients 1500`; (por defecto es 1000). También configuraremos que el orden de las consultas sea aleatorio, así no perderá tiempo en ordenar, con la línea `rrset-order {order random;};`. El archivo `/etc/bind/named.conf.options` quedará por tanto del siguiente modo:



```
named.conf.options
options {
    directory "/var/cache/bind";
    recursive-clients 1500;
    rrset-order {order random;};

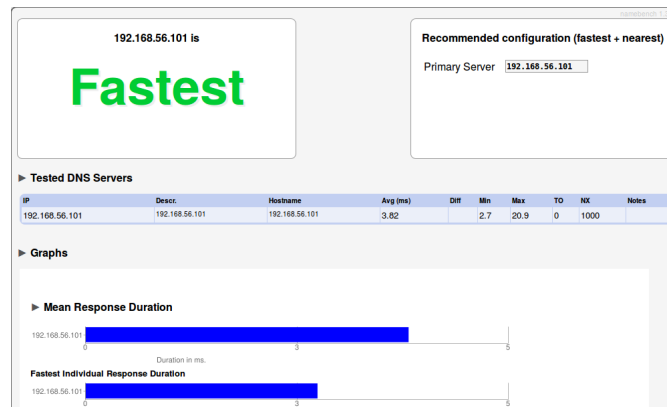
    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

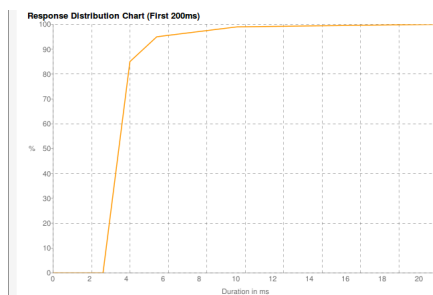
    forwarders {
        8.8.8.8;
    };
}
```

Figura 9.10: Editando `/etc/bind/named.conf.options` para cambiar el rendimiento.

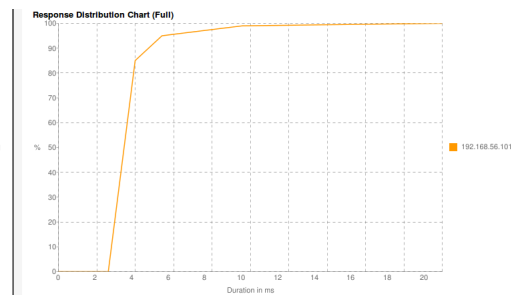
Ahora **reiniciaremos de nuevo el servicio bind9** para que los cambios surtan efecto y ejecutaremos de nuevo el benchmark. Recordemos que el objetivo era reducir el tiempo que de media tarda el servidor en resolver direcciones. Los resultados son los siguientes:



(a) Resultados del benchmark (I).



(b) Resultados del benchmark (II).



(c) Resultados del benchmark (III).

Configuration

Name	Value
benchmark_thread_count	2
enable_sensory_checks	0
health_thread_count	40
health_timeout	3.75
hide_results	0
input_source	firefox
num_servers	11
ping_timeout	0.5
query_count	1000
run_count	1
select_mode	automatic
template	html
timeout	3.5
upload_results	0
version	1.3.1

(d) Resultados del benchmark (IV).

Figura 9.11: Resultados de lanzar DNS Benchmark contra el servidor DNS después de cambiar las opciones.

¡Éxito! Como vemos, hemos conseguido mejorar el rendimiento de forma notable tan sólo cambiando dos parámetros, pues ahora el tiempo medio que tarda en resolver direcciones es de 3.82 ms.

(Fuentes: [14] [4] [13] [7] [16])

Referencias

- [1] Apache.org. Apache mpm worker. <https://httpd.apache.org/docs/2.2/mod/worker.html>.
- [2] Apache.org. Apache performance tuning. <https://httpd.apache.org/docs/2.2/es/misc/perf-tuning.html>.
- [3] Centos.org. Using the systemctl command. https://www.centos.org/docs/5/html/5.1/Deployment_Guide/s1-proc-sysctl.html.
- [4] Conekia Labst. Dns resolucion directa e inversa. <http://labs.conekia.es/dns-resolucion-directa-e-inversa/>.
- [5] Moodle. Recomendaciones sobre desempeño. https://docs.moodle.org/all/es/Recomendaciones_sobre_desempe%C3%B1o#Desempe.C3.B1o_de_Apache.
- [6] NixCraft. Linux increase process identifiers limit with /proc/sys/kernel/pid_max. <http://www.cyberciti.biz/tips/howto-linux-increase-pid-limits.html>.
- [7] Digital Ocean. How to configure bind as a caching or forwarding dns server on ubuntu 14.04. <https://www.digitalocean.com/community/tutorials/how-to-configure-bind-as-a-caching-or-forwarding-dns-server-on-ubuntu-14-04>.
- [8] Norfi PC. Como modificar claves y valores del registro usando la línea de comandos y archivos batch. <https://norfipc.com/comandos/comandos-batch-modificar-registro.html>.
- [9] JGA IT Pro. Windows server 2012 r2 - copia de seguridad al registro de windows regedit. <https://www.youtube.com/watch?v=zam2goffWwc>.
- [10] rm rf. systemctl y /proc/sys – modificar parámetros de kernel. <http://rm-rf.es/sysctl-y-procsys-modificar-parametros-de-kernel/>.
- [11] Microsoft Support. Información del registro de windows para usuarios avanzados. <https://support.microsoft.com/es-es/kb/256986>.
- [12] Microsoft TechNet. Configurar la compresión (iis 7). <https://technet.microsoft.com/es-es/library/cc730629%28v=ws.10%29.aspx>.
- [13] Microsoft TechNet. Descripción de búsqueda inversa. <https://technet.microsoft.com/es-es/library/cc730980.aspx>.
- [14] Webebre. Instalación y configuración de un servidor dns. <http://webebre.net/instalacion-y-configuracion-de-un-servidor-dns/>.
- [15] Wikipedia. Kernel panic. https://es.wikipedia.org/wiki/Kernel_panic.
- [16] Zytrax. Dns bind9 query statements. <http://www.zytrax.com/books/dns/ch7/queries.html>.