Servidores Web de altas Prestaciones

Ejercicios de Teoría

Tema 2 - Alta disponibiliad y escalabilidad

Ejercicio T2.1:

Calcular la disponibilidad del sistema si tenemos dos réplicas de cada elemento (en total 3 elementos en cada subsistema).

Disponibilidades iniciales

Component	Availability
Web	85 %
Application	90 %
Database	99.9 %
DNS	98 %
Firewall	85 %
Switch	99 %
Data Center	99.99 %
ISP	95 %

Con 2 elementos en cada subsistema

Component	Availability
Web	97.75 %
Application	99 %
Database	99.9999 %
DNS	99.96 %
Firewall	97.75 %
Switch	99.99 %
Data Center	99.99 %
ISP	99.75 %

Con 3 elementos en cada subsistema

Component	Availability
Web	99.6625 %
Application	99.9 %
Database	99.9999999 %
DNS	99.9992 %
Firewall	99.6625 %
Switch	99.9999 %
Data Center	99.99 %
ISP	99.9875 %

Ejercicio T2.2:

Buscar frameworks y librerías para diferentes lenguajes que permitan hacer aplicaciones altamente disponibles con relativa facilidad.

- Apache Hadoop: software que permite el procesamiento distribuido de grandes conjuntos de datos
- Microsoft Operations Framework: para productos de alta disponibilidad con tecnología Microsoft.
- Python: Django, TurboGears. Más información
- JavaScript: AJAX
- PHP: CakePHP, Zend, CodeIgniter

Ejercicio T2.3:

¿Cómo analizar el nivel de carga de cada uno de los subsistemas en el servidor?

Todo estos son profilers para monitorizar nuestro servidor (Hardware, software, red, tasa de peticiones...):

- Nagios.
- Munin.
- Zabbix.

Ejercicio T2.4:

Buscar ejemplos de balanceadores software y hardware (productos comerciales).

Software: HaProxy, Nginx, Pound.

Hardware: CISCO, Load Balancer ADC, F5 BIG-IP LTM

Buscar productos comerciales para servidores de aplicaciones.

Ejemplos: GlassFish, Java EE, Sun-Netscape Iplanet.

Buscar productos comerciales para servidores de almacenamiento.

Ejemplos: Dell Compellent FS8600, IBM EXP2500 Storage Enclosure, Windows Azure

Tema 3 - La red de una granja web

Ejercicio T3.1:

Buscar con qué órdenes de terminal o herramientas gráficas podemos configurar bajo Windows y bajo Linux el enrutamiento del tráfico de un servidor para pasar el tráfico desde una subred a otra.

En Linux el comando para pasar el enrutamiendo del tráfico de una subred a otra es route. Con route podemos añadir (add) y eliminar (del) rutas. Además podemos configurar el tráfico de red siguiendo este tutorial de enrutamiento en Linux, utilizando iptables.

Para Windows, en Windows Server podemos añadir un Servicio de enrutamiento y acceso remoto.

Ejercicio T3.2:

Buscar con qué órdenes de terminal o herramientas gráficas podemos configurar bajo Windows y bajo Linux el filtrado y bloqueo de paquetes.

Linux: iptables, ip o tc, ufw.

Windows: Windows Server 2003, route

Herramientas gráficas: Shorewall, WireShark.

Tema 4 - Balanceo de carga

Ejercicio T4.1:

Buscar información sobre cuánto costaría en la actualidad un mainframe. Comparar precio y potencia entre esa máquina y una granja web de unas prestaciones similares.

> "Mainframe prices start at around $40,000 for an entry-level system. While Server farm can be as big or small, what you need " - www.suse.com

Los mainframes tienen el problema de que pueden no ajustarse a nuestras necesidades mientras que la granja web se configurará según nuestro criterio.

Ejercicio T4.2:

Buscar información sobre precio y características de balanceadores hardware específicos. Compara las prestaciones que ofrecen unos y otros.

KEMP LM-5400 que rondaría unos 19000$

Citrix MPX-8005 que rondaría los 24000$

En cuánto a características tendríamos lo siguiente:

-Número de fuentes de alimentación: Citrix(1) y KEMP(2).

-Consumo(medido en W): Citrix(184.5) y KEMP(185). Sería muy similar

-Memoria RAM:Citrix(32GB) y KEMP(8GB)

Ejercicio T4.3:

Buscar información sobre los métodos de balanceo que implementan los dispositivos recogidos en el ejercicio 4.2

Citrix sobre todo emplea Round Robin o Least Connection.

KEMP LM-5400 implementa los algoritmos de Round Robin, Round Robin con pesos, el menor número de conexiones, menor número de conexiones con pesos, reparto mediante un sistema agente que estudia la carga del servidor, source o ip-hash, switching según el contenido de la capa 6 y usar un servidor por defecto y usar otro cuando haya mucha demanda.

Ejercicio T4.5:

Probar las diferentes maneras de redirección HTTP. ¿Cuál es adecuada y cuál no lo es para hacer balanceo de carga global? ¿Por qué?

- Redirección básica con HTML
- Redirección con PHP
- Redirección con JavaScript

Todos ellos tienen el mismo problema y es que tienen un único punto de fallo y es el servidor principal que hace la redirección a los distintos centros de datos. La única forma adecuada de hacer balanceo de carga global es mediante GSLB

Ejercicio T4.6:

Buscar información sobre los bloques de IP para los distintos países o continentes.

“http://services.ce3c.be/ciprg/” nos proporicona listas en función del pais seleccionado

```
<rule address="2.136.0.0" action="deny" type="address" mask="255.248.0.0"
comment="Spain" />
<rule address="2.152.0.0" action="deny" type="address" mask="255.252.0.0"
comment="Spain" />
<rule address="5.1.32.0" action="deny" type="address" mask="255.255.248.0"
comment="Spain" />
<rule address="5.2.24.0" action="deny" type="address" mask="255.255.248.0"
comment="Spain" />

...
<rule address="217.172.64.0" action="deny" type="address" mask="255.255.240.0"
comment="Spain" />
<rule address="217.173.112.0" action="deny" type="address" mask="255.255.240.0"
comment="Spain" />
<rule address="217.197.16.0" action="deny" type="address" mask="255.255.240.0"
comment="Spain" />
<rule address="217.198.192.0" action="deny" type="address" mask="255.255.240.0"
comment="Spain" />
<rule address="217.216.0.0" action="deny" type="address" mask="255.254.0.0"
comment="Spain" />
```

Bastante extensa la lista.

Ejercicio T4.7:

Buscar información sobre métodos y herramientas para implementar GSLB.

Deberíamos colocar dos servidores en zonas distintas, una vez hecho esto, crearíamos una VPN para conectar de manera segura ambos servidores previamente instalados. Después solo habría que configurar cada balanceador de forma eficiente para que reparta el tráfico.

Tema 5 - Medir prestaciones

Ejercicio T5.1:

Buscar información sobre cómo calcular el número de conexiones por segundo.

En Linux bastaría con ejecutar el siguiente comando: “netstat | grep http | wc -l”, nos devuelve con un entero que representa cada cliente que posee sockets para atender diferentes peticiones.

También hay otras aplicaciones de gran utilidad como pueden ser ipstate o apache2ctl,además de la mencionada anteriormente netstat

Ejercicio T5.3:

Buscar información sobre características, disponibilidad para diversos SO, etc de herramientas para monitorizar las prestaciones de un servidor. Para empezar, podemos comenzar utilizando las clásicas de Linux:

- *top : Software que nos proporciona información acerca de los procesos en ejecución además de el tiempo de CPU/RAM empleados en cada uno de ellos.*
- *Vmstat : Software que sirve para informar sobre las estadísticas de memoria virtual y proporcionar información sobre eventos del sistema, como carga de CPU, paginación, número de cambios de contexto, interrupciones de dispositivo y llamadas del sistema. El comando* `vmstat` *también puede mostrar las estadísticas de intercambio, vaciado de memoria caché e interrupciones.*
- *Netstat : Software que nos da información de todos aquellos sockets abiertos por cada proceso en ejecución, además de información como puerto origen y destino, ip origen y destino, etc.*

Monitorización a gran escala:

- Pandora FMS: *Versión libre capaz de monitorizar más de 10,000 nodos y cubre (sin limitaciones) una monitorización de red, de servidores (basados en agentes o de forma remota) y de aplicaciones. Con funcionalidades completas de informes, alertas, integraciones con terceros via API.*
- *Zabbix: Fácil configuración y potente interfaz gráfico. Empieza a caer su rendimiento cuando se empiezan a monitorizar muchos nodos (A partir de 10,000 nodos).* Tema 6 - Asegurar el sistema web

Ejercicio T6.1:

Aplicar con iptables una política de denegar todo el tráfico en una de las máquinas de prácticas.

Comprobar el funcionamiento.

```
# Política por defecto: denegar todo el tráfico
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

$> ping www.google.es

Tiempo de espera agotado para esta solicitud.

Tiempo de espera agotado para esta solicitud.

Aplicar con iptables una política de permitir todo el tráfico en una de las máquinas de prácticas. Comprobar el funcionamiento.

```
# Política por defecto: denegar todo el tráfico
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
```

Ejercicio T6.2:

Comprobar qué puertos tienen abiertos nuestras máquinas, su estado, y qué programa o demonio lo ocupa

Conexiones activas

Proto	Dirección local	Dirección remota	Estado
TCP	0.0.0.0:135	DESKTOP-JLFHT9H:0	LISTENING
TCP	0.0.0.0:445	DESKTOP-JLFHT9H:0	LISTENING
TCP	0.0.0.0:5040	DESKTOP-JLFHT9H:0	LISTENING
TCP	0.0.0.0:7680	DESKTOP-JLFHT9H:0	LISTENING
TCP	0.0.0.0:8884	DESKTOP-JLFHT9H:0	LISTENING
TCP	127.0.0.1:49873	DESKTOP-JLFHT9H:49880	ESTABLISHED

….

Ejercicio T6.3:

Buscar información acerca de los tipos de ataques más comunes en servidores web (p.ej. secuestros de sesión). Detallar en qué consisten, y cómo se pueden evitar.

-Los ataques de inyección de código SQL, es una técnica para modificar una cadena de consulta de base de datos mediante la inyección de código en la consulta.

-La Denegación de Servicio (DoS) es la forma más comunes para congelar el funcionamiento de un sitio web. Los ataques de denegación de servicio por lo general se dirigen a puertos específicos, rangos de IP o redes completas, pero se pueden dirigir a cualquier dispositivo o servicio conectado.

-Los atacantes utilizan Cross-site Scripting (XSS) para inyectar scripts maliciosos en lo que serían sitios web inofensivos. El XSS generalmente se utiliza para obtener acceso de la cuenta de un usuario.

Tema 7 - Almacenamiento de datos

Ejercicio T7.1:

- *¿Qué tamaño de unidad de unidad RAID se obtendrá al configurar un RAID 0 a partir de dos discos de 100 GB y 100 GB?*

Se obtendrá una unidad de 200GB

- *¿Qué tamaño de unidad de unidad RAID se obtendrá al configurar un RAID 0 a partir de tres discos de 200 GB cada uno?*

Se obtendrá una unidad de 600GB

Ejercicio T7.2:

- *¿Qué tamaño de unidad de unidad RAID se obtendrá al configurar un RAID 1 a partir de dos discos de 100 GB y 100 GB?*

Se obtendrá una unidad de 100GB

- *¿Qué tamaño de unidad de unidad RAID se obtendrá al configurar un RAID 1 a partir de tres discos de 200 GB cada uno?*

Se obtendrá una unidad de 200GB