

Moulay Rchid Drissi Maazouz  
Nicolás García Muñoz

# **Balanceadores de carga y algoritmos de balanceo**

## 1.Introducción

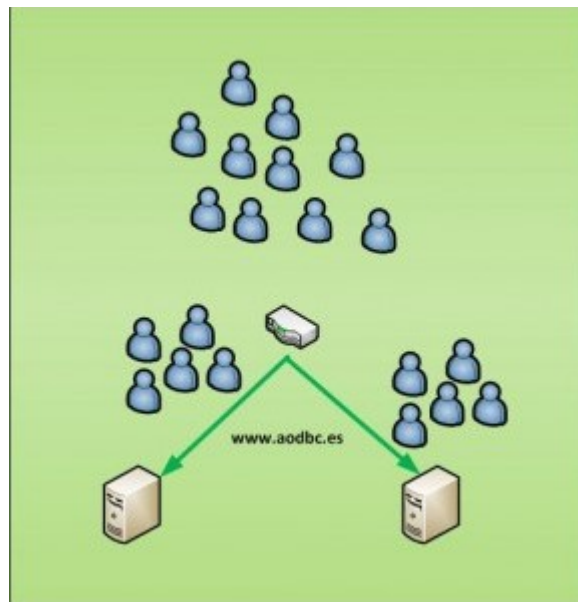
Cuando diseñamos una arquitectura de red para dar soporte a uno o varios servicios, una de las características más importantes junto a la seguridad, tolerancia a fallos y flexibilidad, será la escalabilidad.

En este Trabajo detallamos dos balanceadores de carga software de código abierto , explicamos detalladamente su instalación, configuración y uso. Por último, en ellos se realizarán mediciones para visualizar los distintos resultados obtenidos mediante diferentes algoritmos de balanceo.

## 2.Balanceador de carga software

### 2.1 zenloadbalancer

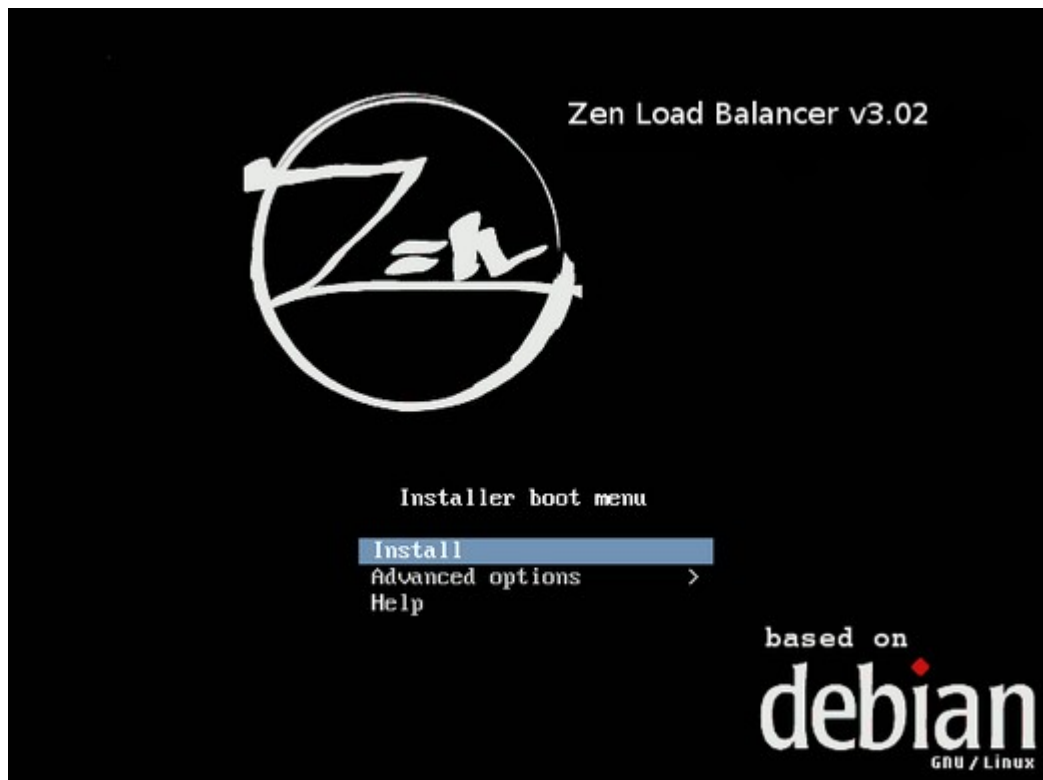
Zen es un proyecto de equilibrador de carga de código abierto que proporciona un conjunto completo de herramientas para ejecutar y administrar una solución completa de equilibrador de carga que incluye: granja y definición de servidor, red, el agrupamiento, el seguimiento, la gestión de los certificados de seguro, registros, copias de seguridad de configuración, enlace ascendente balanceo de carga de apoyo, y mucho más.



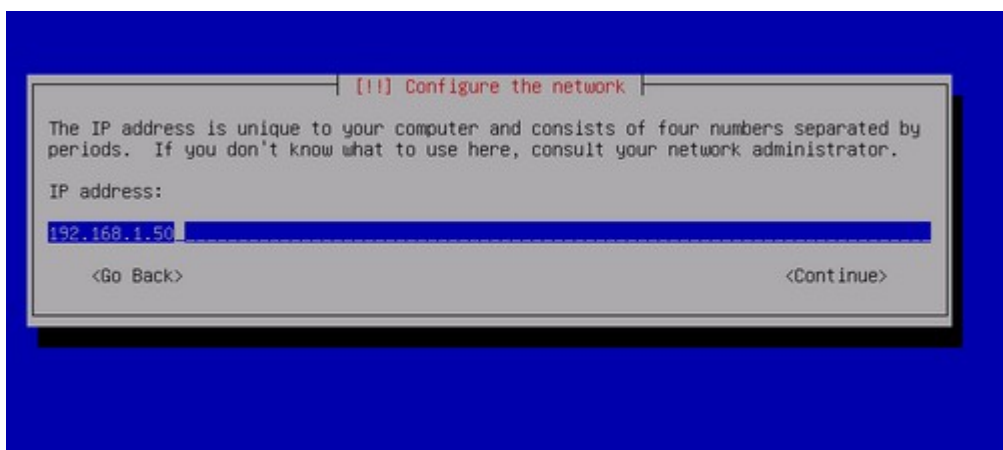
## 2.2 Características

- Balanceo de carga
- Equilibrador de carga de red
- Alta disponibilidad
- Open Source equilibrador de carga
- Equilibrador de carga de TCP
- Equilibrador de carga Web
- Equilibrador de carga UDP
- Equilibrador de carga SSL
- Equilibrador de carga SIP
- Equilibrador de carga FTP
- Equilibrador de carga TFTP

## 2.3 Instalación



A continuación, el proceso de instalación va a configurar la interfaz de red, es necesario configurar una dirección IP estática que va a ser utilizado en el inicio para acceder al panel de administración Web Zen. Se solicitará información de configuración adicional, como la máscara de red, puerta de enlace y el servidor DNS a lo largo del proceso de instalación.



### 3.- Configuración de Zenloadbalancer mediante la interfaz

#### 3.1 crear Farm

En la pestaña Manage → Farms, se establecerá la maquina balanceadora así

#### Manage::Farms

Farms table									
Name	Virtual IP	Virtual Port(s)	Pending Conns	Established Conns	Closed Conns	PID	Status	Profile	Actions
GranjaZenload	172.16.76.142	80	0	0	0	1294		tcp	

**Pending Conns:** se calculan con la entrada de solicitudes SYN que están pendientes de ser procesados en el sistema de esta granja.

**The Established Conns:** se calculan con la entrada ESTABLECIDO peticiones que se están procesando actualmente.














**The Closed Conns:** se calculan con las conexiones ESPERA CERRAR que ya han sido tratados en el sistema.

**Protocol Type:TCP:** Al activar esta opción, la granja está escuchando por conexiones TCP entrantes al IP actual virtual

### 3.2 Edición de los servidores granja real







Una vez que se crea una nueva comunidad, tenemos que incluir los servidores con los servicios reales con el fin de entregar las conexiones de cliente.

Edit real IP servers configuration

Server	Address	Port	Max connections	Weight	Priority	Actions
0	172.16.76.140	80	1000	0	0	  
1	172.16.76.130	80	1000	0	0	  
2	172.16.76.137	80	1000	0	0	  
3	172.16.76.138	80	1000	0	0	  
						

Cancel

Farms table

Name	Virtual IP	Virtual Port(s)	Pending Conns	Established Conns	Closed Conns	PID	Status	Profile	Actions
GranjaZenload	172.16.76.142	80	0	0	0	1294		tcp	   
									

**Max connections:** Es el número máximo de conexiones simultáneas que el servidor real actual será capaz de recibir. Este valor debe ser inferior a los clientes máximos de los parámetros globales.

### 3.3 Editar parámetros globales GranjaZenload

## Edit GranjaZenload Farm global parameters

**Farm's name** \*service will be restarted.

**Load Balance Algorithm.**

☒ **Enable client ip address persistence through memory.**

**Max number of clients memorized in the farm** \*service will be restarted.

**client-time(sec, 0=always)**

**Backend response timeout secs.**

**Max number of simultaneous connections that manage in Virtual IP** \*service will be restarted.

**Max number of real ip servers** \*service will be restarted.

☐ **Add X-Forwarded-For header to http requests.**

**Frequency to check resurrected backends secs.**

☐ **Use FarmGuardian to check Backend Servers.**

**número máximo de clientes memorizados en la granja:** Estos valores sólo tienen sentido si se habilita la persistencia IP del cliente. El campo de cliente es sobre el número máximo de clientes que serán posibles de memorizar y el valor de tiempo es el tiempo máximo de vida de esta clientes a memorizar

**Tiempo de espera de respuesta del servidor:** Es el segundo máximo que el servidor real tiene que responder a una solicitud. Si la respuesta backend es demasiado tarde, entonces el servidor se marcará como una lista negra.

**número máximo de conexiones simultáneas para la IP virtual.** Es el valor máximo de conexiones establecidas y los clientes activos que el servicio virtual será capaz de gestionar

**número máximo de servidores IP reales:** Es el número máximo de servidores reales que la granja será capaz de haber configurado.

**Frecuencia para comprobar backends resucitados:** Este valor en segundos es el tiempo para salir un servidor real en la lista negra y comprueba si está vivo.

**Utilice farmguardian para comprobar los servidores back-end:** Al marcar esta casilla permitirá a un estado de vigilancia más avanzado para backends y totalmente personalizado para sus propios scripts. Cuando se detecta un problema por farmguardian desactiva automáticamente el servidor real y se marcarán como la lista negra.

**Load Balance Algorithm:** Por defecto, el algoritmo de peso será el algoritmo predeterminado seleccionado.

**Round Robin :** algoritmo de balanceo basado en una cola de turnos. Adecuado si todos tienen potencia similar.

**Menor número de conexiones ( Least Connections ):** se reparte el trabajo según la utilización de cada una ( el balanceador lleva la cuenta de cada uno ). Adecuado para distribuir el trabajo entre máquinas similares, evitándose la sobrecarga de las que puedan tener mas trabajo.

**Ponderación ( Weight ) :** se reparte trabajo según un peso asignado a cada máquina. Adecuado para cuando los dos anteriores algoritmos no den buen resultado, es necesario ajustarlo según las capacidades de las máquinas.

*Es el valor del peso para el servidor real actual, que sólo es útil si el peso algoritmo está habilitado. Más valor de peso indica que hay más conexiones entregados al programa de fondo actual. Por defecto se establecerá un valor de peso de 1.*

**Prioridad:** similar a la ponderación pero con grupos de máquinas. Cada grupo tiene una prioridad y un máximo de conexiones simultaneas que puede atender. *Es el valor de prioridad para el servidor real actual, que sólo es útil si se habilita el algoritmo de prioridad. El valor de prioridad aceptado es entre 1 y 9, de menos valor indica mayor prioridad al servidor real actual. Por defecto se establecerá*

*un valor de prioridad 1.*

**Tiempo de respuesta :** se reparte el trabajo según los tiempos de respuesta de cada máquina. El balanceador debe de estar monitorizando las máquinas.

**Combinación :** se reparte el trabajo usando una mezcla de los algoritmos anteriores. Es el mas eficientes. Por ejemplo, ZenLoad usa un algoritmo basado en tiempo de respuesta y menor número de conexiones.

### ***Persistence Mode:***

**No persistencia:** La granja no usará ningún tipo de persistencia entre el cliente y el servidor.

**la persistencia de IP:** Al activar esta opción, la granja asignará el mismo backend para cada conexión con respecto a la dirección IP de origen del cliente.

**Source IP Address Persistence time to live:** Si se selecciona cualquier persistencia, este valor de campo indica el número de segundos que se le ha asignado la persistencia entre la fuente de cliente y el servidor.

**Use farmguardian to check backend servers:** Al marcar esta casilla permitirá a un estado de vigilancia más avanzado para backends y totalmente personalizado con sus propios scripts. Cuando un mal comportamiento es detectado por farmguardian, se desactiva automáticamente el servidor real y se marcará como FGdown. Este es un servicio independiente por lo que no tiene necesidad de reiniciar el servicio de granja.

## ***3.4 Experimento y Resultados***

si no obtenemos ningún mensaje de error, todo está funcionando correctamente ya podemos probar la configuración haciendo peticiones a **la IP** de esta máquina balanceador. Por ejemplo, podemos usar **el comando curl** de la siguiente forma:

```
curl http://172.16.76.142
```

```
curl http://172.16.76.142
```

*Ver Resultados*














## ***4 Aplicar otro algoritmo de ponderación***

***Establecemos pesos de:***








Balanceador zenload    RAM:952MB  
 Maquina1: RAM 1024MB    peso : 2  
 Maquina2: RAM 2048MB    peso : 4  
 Maquina3: RAM 1200MB    peso : 2  
 Maquina4:RAM 3072MB    peso : 6

Edit real IP servers configuration

Server	Address	Port	Max connections	Weight	Priority	Actions
0	172.16.76.140	80	6	2	0	  
1	172.16.76.130	80	6	4	0	  
2	172.16.76.137	80	6	2	0	  
3	172.16.76.138	80	6	6	0	  
						

Cancel

Farms table

Name	Virtual IP	Virtual Port(s)	Pending Conns	Established Conns	Closed Conns	PID	Status	Profile	Actions
GranjaZenload	172.16.76.142	80	0	0	0	1294		tcp	  
									

## 4.1 Experimento y Resultados

## 5 Bibliografía

<https://www.zenloadbalancer.com> (Para instalar zenloadbalancer)

<http://www.driverlandia.com/configurar-tarjeta-de-red-con-ip-estatica-en-debian-sin-interfaz-grafica/> (Para configurar la red)

<https://www.zenloadbalancer.com/zlb-administration-guide-v304/#prettyPhoto>

