

NAME

balance 3.54 – Simple proxy TCP con mecanismos de balanceo de carga y conmutación por error.

SYNOPSIS

balance [**-b** *addr*] [**-B** *addr*] [**-t** *sec*] [**-T** *sec*] [**-adfpHM6**] port host1[:port1[:maxc]] [!%] [... hostn[:portn[:maxc]]]

balance [**-b** *addr*] **-i** [**-d**] [**-M**] port

balance [**-b** *addr*] **-c** *cmd* [**-d**] [**-M**] port

DESCRIPTION

Balance es un simple proxy TCP fácil de usar que permite el uso de balanceo de carga mediante round-robin o hash. Ofreciendo también conmutación por error entre varios destinos.

Balance soporta IPv6 a la escucha de peticiones para balancear, lo cual lo hace muy útil en migraciones de servicios y servidores de IPv4 a IPv6.

Balance está disponible en <http://balance.sourceforge.net>.

Definición: Un destino compuesto por la dirección de un host y un puerto se denomina "canal". Un canal a su vez es miembro de un grupo de canales. Los canales son numerados dentro de un grupo empezando por el 0. Los grupos de canales a su vez son también numerados a partir de 0, siendo este el grupo inicial por defecto.

Balance acepta conexiones en el puerto configurado y las envía a los canales correspondientes. Al menos un canal (en el grupo por defecto) debe ser especificado. Si hay dos o más canales dentro de un grupo, *balance* realiza un simple balanceo por round-robin (por defecto) o hash entre los canales que estén disponibles.

Balance permite la definición de nuevos grupos de canales. El esquema de conexión funciona de la siguiente manera: *balance* trata de establecer conexiones a los canales del primer grupo (0) y realiza el balanceo de carga entre ellos. Si no hay un canal disponible en el primer grupo, *balance* prosigue ascendente-mente con el siguiente grupo de canales. Los grupos son siempre separados mediante el carácter "!" en la línea de comandos y pueden ser controlados dentro del modo interactivo con el comando "group".

Utilizar "%" en vez de "!" como separador de grupo, indica que el balanceo para el grupo previo es del tipo "hash". A diferencia del algoritmo de round-robin, una distribución por hash se basa en la dirección IP del cliente, la cual determina el canal destino. Esto permite conectar clientes siempre el mismo servidor (ej. balanceo en sesiones http).

Los hosts destino pueden ser especificados usando el hostname o la IP. Los puertos pueden ser nombrados mediante su número, o bien el nombre de servicio utilizando /etc/services. Si no se especifica un puerto destino se usará de manera predeterminada el mismo puerto origen que *balance* esté manejando en dicho canal.

Balance permite configurar un límite máximo de conexiones por canal. Este parámetro puede ser añadido justo después de especificar el puerto mediante dos puntos (":"). Si se especifica un número máximo de conexiones en un canal, este sólo será utilizado hasta alcanzar dicho máximo de conexiones simultáneas. Un valor de 0 en "maxc" indica que no hay límite definido, siendo esa la configuración predeterminada.

El número máximo de grupos y canales soportados en *balance* puede ser configurado en tiempo de compilación, teniendo de forma predeterminada un límite de 16 canales en 16 grupos.

La conmutación por error a otro destino ("canal") ocurre cuando la conexión es denegada en el canal en uso, o bien si el tiempo de espera tratando de establecer una conexión vence. Si todos los canales fallan, la conexión del cliente con el balanceador será cerrada.

Balance acepta las siguientes opciones:

- 6** Fuerza conexiones al socket IPv6 usando hints.ai_family (AF_INET6).
- a** Habilita la opción "autodisable": El canal debe ser reactivado manualmente usando el modo interactivo después de una falla.

- b** Bindhost: *Balance* configura una dirección específica para escuchar peticiones en vez de INADDR_ANY.
- B** Bindhost: *Balance* define un host o dirección IP en concreto para conexiones salientes (la conexión será inicializada desde dicha dirección).
- c** Command: Permite enviar instrucciones al proceso principal de balance (ver modo interactivo).
- d** Debug: *Balance* muestra en la salida estándar los mensajes en modo depuración.
- H** Hashfailover: *Balance* realiza conmutación por error aunque se esté usando balanceo por hash.
- F** Foreground: obliga a *balance* a permanecer en primer plano. Útil para pruebas o procesos de depuración. *balance* puede ser parado en dicho modo utilizando ^C (u otro carácter de interrupción).
- M** En vez de usar memoria compartida, utiliza mapeos de memoria IPC.
- i** Control Interactivo: *Balance* puede mediante el uso de memoria compartida conectar con el proceso en ejecución, el cual es definido por la dirección origen y puerto utilizado. Esto permite controlar su configuración en tiempo de ejecución mediante una interfaz de línea de comandos. El permiso de acceso mediante esta interfaz se determina por las restricciones de acceso que tenga el segmento de memoria compartida en vigor. *help* o *?* muestra una versión resumida de las opciones disponibles, *assign* permite reasignar un canal dentro de un grupo (sólo si está desactivado), *create* crea una nueva dirección destino y puerto (canal) en el grupo actual. *disable* deshabilita el canal en el grupo actual, *enable* habilita un canal previamente deshabilitado en el grupo actual, *group* cambia el grupo en vigor por otro sin salir del modo interactivo *hash* establece el tipo de balanceo Hash al grupo, *help* muestra la ayuda, *kill* mata el proceso padre y sale del modo interactivo, *maxc* <channel> <maxc> fija un número máximo de conexiones a un canal (0 equivale a ilimitado), *mrtg-bytes* <group> <channel> visualiza el tráfico enviado y recibido en un formato compatible con MRTG (destinado a ser llamado con -c de forma automática por MRTG). *mrtg-conns* <group> <channel> muestra el total de conexiones en formato compatible con MRTG (destinado a ser llamado con -c automáticamente por MRTG). *quit* sale del modo interactivo, *reset* reinicia los contadores de bytes del canal especificado, *rr* cambia el tipo de balanceo de carga del grupo por el de "Round Robin", *show* muestra una visión general de la transferencia de bytes de entrada y salida de todos los canales. La salida es ordenada por grupos. También son visualizadas las conexiones actuales (c) y sus límites máximos permitidos (maxc), *version* muestra la versión y las constantes MAXGROUPS y MAXCHANNELS utilizadas en la compilación.
- p** Packetdump: *Balance* muestra todos los datos entrantes y salientes por la salida estándar utilizando una representación sencilla y legible. Útil para tareas de depuración y análisis de protocolo.
- t** Connect Timeout: modifica el tiempo de espera predeterminado para establecer una conexión a cualquier destino. El tiempo de espera para considerar una conexión inaccesible es de 5 segundos.
- T** Select Timeout: Tiempo de espera para select(), default = 0 (nunca). Esta opción es todavía experimental.

EJEMPLOS

\$ balance smtp host1.test.net host2.test.net

Conecta con el puerto SMTP local y reenvía el tráfico alternando entre el puerto SMTP de host1 y host2. *Balance* arranca automáticamente en segundo plano.

\$ balance -b 2001:DB8::1 80 10.1.1.1 10.1.1.2

Balance Escucha en el puerto 80 de la dirección local IPv6 2001:DB8::1 y distribuye las conexiones entre las direcciones IPv4 10.1.1.1 y 10.1.1.2.

\$ balance -b ::ffff:10.1.1.3 80 10.1.1.1 10.1.1.2

Balance Escucha en el puerto 80 de la dirección IPv4 10.1.1.3 (en notación IPv6) y distribuye las conexiones a las direcciones IPv4 10.1.1.1 y 10.1.1.2.

\$ balance -fp imap mailserver

Conecta con el puerto local IMAP y reenvía todo el tráfico al host "mailserver". *Balance* permanece en primer plano mostrando en la salida estándar información legible.

\$ balance -f 8888 host1 10.1.1.1:8000

Escucha en el puerto local 8888 y reenvía el tráfico alternando entre el host1 y su puerto 8888 y el puerto 8000 del host 10.1.1.1. *Balance* permanece en primer plano conectado a la tty.

\$ balance imap mailserver1::16 ! mailserver2

Dos grupos son especificados, cada uno contiene un único canal. El primero puede manejar 16 conexiones simultaneas reenviando información a mailserver1. Tan pronto como ese límite de conexiones es sobrepasado, *balance* procede a utilizar el siguiente grupo de canales reenviando el tráfico al puerto imap del host mailserver2.

\$ balance pop3 host1 host2 host3 ! failover1

Balance hace un balanceo mediante round robin para los tres hosts del grupo 0 usando el puerto del servicio pop3. Si los tres hosts no están disponibles, se conmutará por error al host "failover1".

\$ balance telnet target.munich.net::1

Aquí *balance* limita a una todas las conexiones al puerto telnet local.

\$ balance 8888 localhost::12 ! localhost::4 ! localhost::2 localhost::2 ! localhost:25

Esta es una prueba formando 5 grupos donde se redirecciona el tráfico a si mismo 20 veces. Es simplemente un test que se puede realizar en casa.

BUGS

En caso de que *balance* no pueda enviar una conexión a cualquier destino, la conexión inicial para balancear es aceptada y acto seguido se cierra inmediatamente. Esto no es siempre percibido por el host destino.

AUTHOR

Thomas Obermair, Inlab Software GmbH (obermair@acm.org)

Copyright (c) 2000-2009,2010 por Thomas Obermair (obermair@acm.org) y Inlab Software GmbH (<http://www.inlab.de>), Gruenwald, Alemania. Todos los derechos reservados.

Esta página de manual está traducida y mantenida por Busindre (busilezas@gmail.com).

Balance está publicado bajo la licencia GNU GENERAL PUBLIC LICENSE, leer el fichero COPYING en el paquete de código fuente.