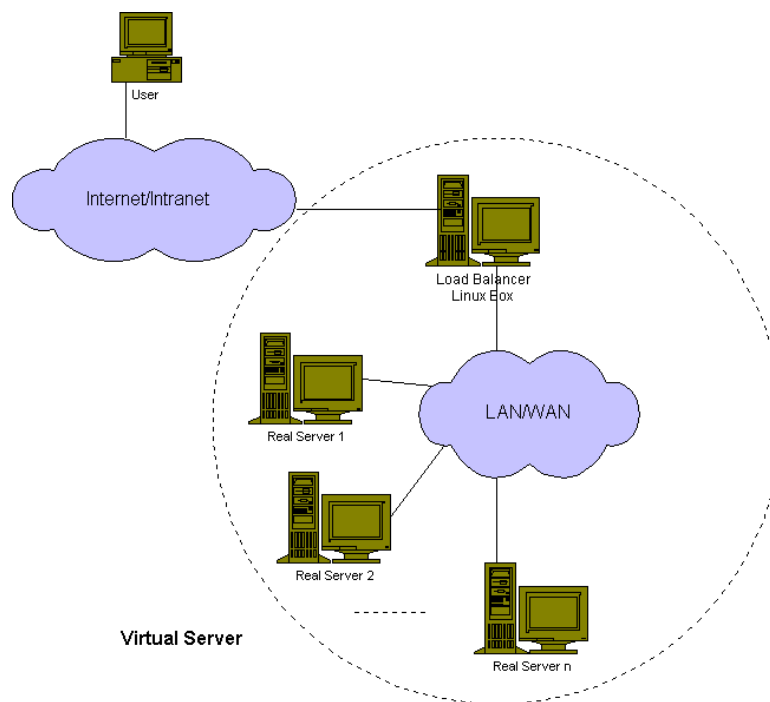


Linux Virtual Server (LVS).

Virtual Server (VS) es un sistema de alta disponibilidad y altamente escalable para manejar un cluster de servidores reales. Por diseño, la arquitectura del “servidor de cluster” es transparente para el usuario final, lo que permite a estos interactuar con él como si se tratara de un solo servidor.



Los servidores reales y el balanceador pueden estar conectados por una red de alta velocidad (Ethernet, Wifi, etc.) o bien dispersos en una WAN. Los balanceadores de carga despachan las peticiones a los diferentes servidores y hacen que los servicios paralelos del cluster aparezcan como un solo servidor virtual con una sola dirección IP. La escalabilidad del sistema se alcanza gracias a la transparencia al momento de agregar o quitar nodos al cluster. La alta disponibilidad es proporcionada por la detección de nodos o bien por fallas de algún demonio, y la reconfiguración del sistema de forma adecuada.

Funcionamiento del LVS.

Actualmente, LVS puede ser implementado de 3 formas diferentes, usando las 3 técnicas de balanceo IP (forwarding de paquetes) existentes en el LinuxDirector:

- Virtual Server vía NAT
- Virtual Server vía IP Tunneling
- Virtual Server vía ruta directa (algo así como enrutamiento estático).

Virtual Server vía NAT.

La ventaja de usar VS vía NAT es que los servidores reales pueden usar cualquier sistema operativo que soporte TCP/IP, usando direcciones privadas (RFC 1819), ya que solo una dirección IP es requerida para balanceo de carga.

Una desventaja al usar este diseño es que la escalabilidad está limitada. El encargado de balancear la carga puede hacer ser un cuello de botella para todo el sistema cuando el número de nodos (servidores o PC) es de 20+, ya que la petición y la respuesta de paquetes necesitan ser escritas por el balanceador de carga.

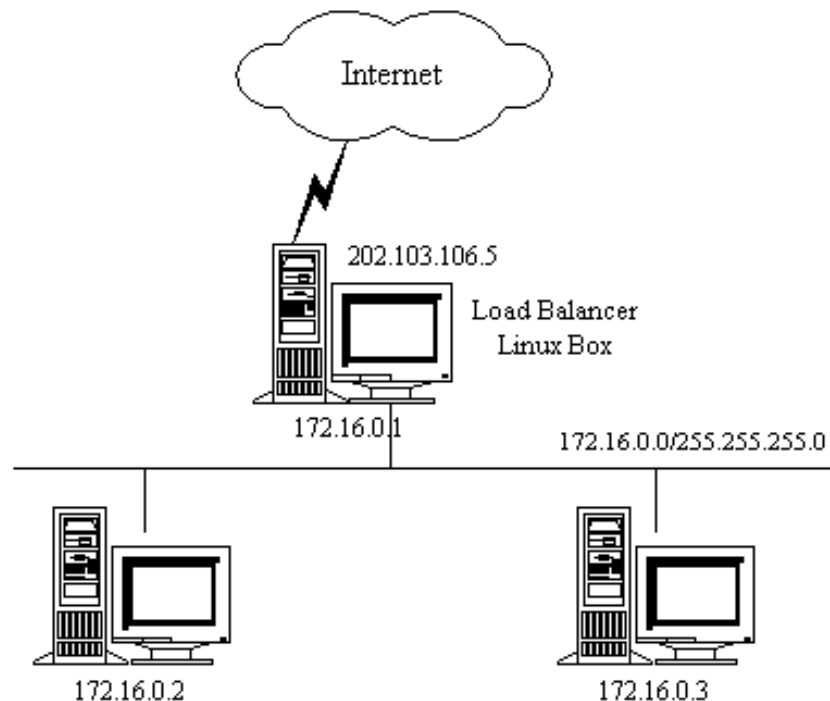
VS vía NAT permite aprovechar de mejor manera el desempeño de varios servidores. Incluso cuando el balanceador de carga comienza a ser un cuello de botella para todo el sistema, existen dos formas de resolverlo. La primera es un acercamiento híbrido, y la otra es usa VS vía IP Tunneling o VS vía ruta directa.

En el acercamiento híbrido existen varios encargados del balanceo de carga los cuales tendrán su propio cluster de servidores, y los balanceadores de carga serán agrupados en un solo dominio llamado Round-Round DNS.

Funcionamiento.

VS vía NAT sobre Linux se realiza usando traducción de puertos, lo que Cisco nos enseña como PAT.

Consideremos la siguiente figura:



*Notemos que los servidores reales corren cualquier Sistema Operativo que soporte TCP/IP. El router por defecto **debe** ser el Virtual Server.*

En esta situación, todo el tráfico que venga desde Internet hacia la dirección 200.103.106.5:80 será “balanceada” a la dirección 172.16.0.2:80 y a 172.16.0.3:8000, mientras que todo el tráfico que venga desde Internet hacia la dirección 200.103.106.5:21 será “balanceada” a la dirección 172.16.0.3:21.

La re-escritura de paquetes funcionará de la siguiente manera:

Paquete de entrada para servicios Web tendrá como origen y destino las siguientes direcciones:

Origen	200.103.190.45:3456	Destino	200.103.106.105:80
---------------	----------------------------	----------------	---------------------------

Ahora bien, el balanceador de carga elegirá un servidor real (de su LAN), por ejemplo 172.16.0.3:8000. Entonces el paquete será re-escrito y enviado (forwarded) al servidor como:

Origen	200.103.190.45:3456	Destino	172.16.0.3:8000
---------------	----------------------------	----------------	------------------------

Y la respuesta que tendrá el balanceador de carga será:

Origen	172.16.0.3:8000	Destino	200.103.190.45:3456
---------------	------------------------	----------------	----------------------------

Finalmente el paquete será re-escrito antes de ser devuelto al cliente, y quedará de la siguiente forma:

Origen	200.103.106.5:80	Destino	200.103.190.45:3456
---------------	-------------------------	----------------	----------------------------

Lo cual no demuestra la transparencia del sistema para el usuario final.

Virtual Server vía IP Tunneling.

En la implementación vista anteriormente (vía NAT), las peticiones y respuesta de todos los paquetes pasaban necesariamente por el balanceador de carga el cual producía un cuello de botella cuando el número de nodos servidores era 20+, ya que el rendimiento de procesamiento de la interfaz de red está eventualmente limitada. Muchas veces hemos visto en servidores de Web que la petición de paquetes es usualmente corta mientras que la respuesta tarda un buen rato...

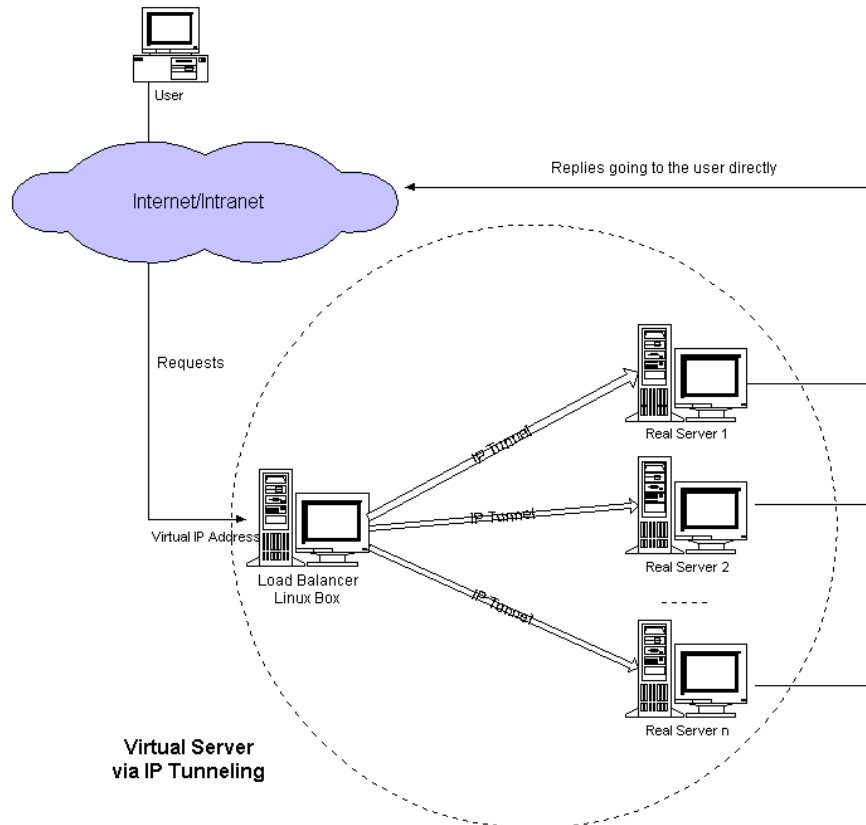
En el VS vía IP Tunneling, el balanceador de carga solo programa las peticiones a los diferentes servidores reales, y son estos servidores reales los que envían las respuestas directamente a los usuarios finales. Entonces, el balanceador de carga puede manejar un gran número de peticiones, puede programar 100+ servidores reales y no será un cuello de botella.

En este caso, todos los servidores deben tener en protocolo “IP Tunneling” (encapsulación IP) habilitado.

Funcionamiento.

IP Tunneling (encapsulación IP) es una técnica para encapsular datagramas UP dentro de datagramas IP, lo que permite a datagramas destinados por una IP ser envueltos y redireccionados a otra dirección IP.

Consideremos la siguiente figura:



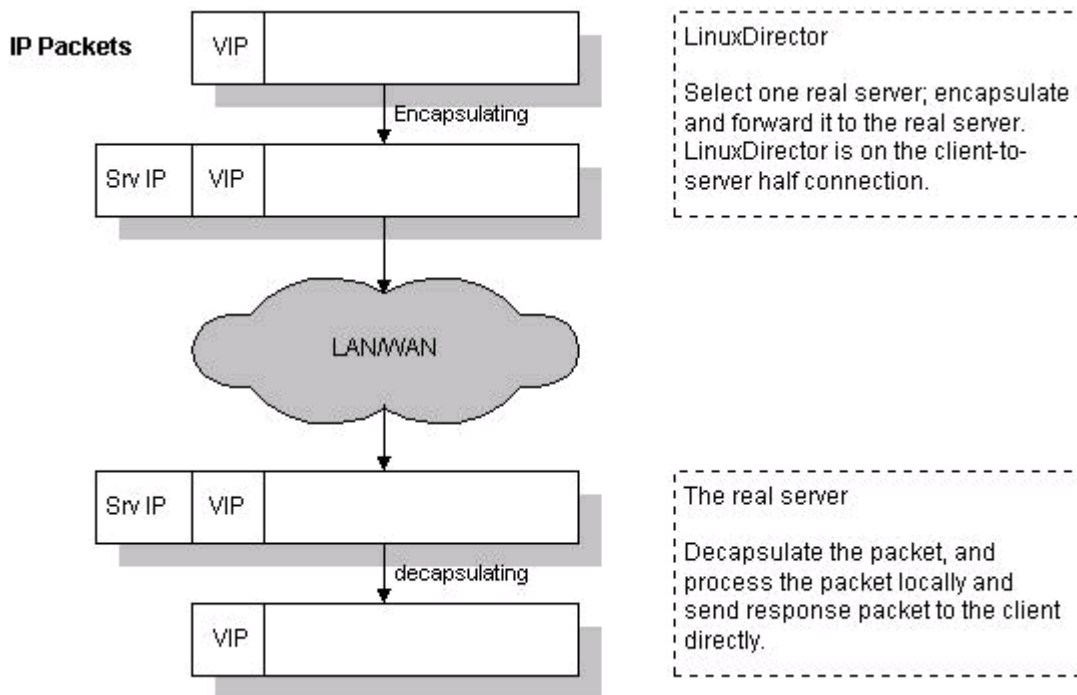
Cuando un usuario accede a un servicio virtual proporcionado por el cluster, llega un paquete destinado a la IP virtual (la IP para el servidor virtual). El balanceador de carga examinará la dirección destino y el puerto del paquete. Si existe un “match” con un servicio virtual, el servidor real será elegido de acuerdo a un algoritmo de programación de la conexión, y la conexión se agregará a la tabla “hash”.

Luego de esto, el balanceador de carga encapsulará el paquete entrante dentro de un datagrama IP y se lo enviará al servidor elegido.

Cuando un paquete entrante perteneciente a esta conexión y al servidor elegido aparece en la tabla “hash”, será encapsulado nuevamente y reenviado a ese servidor. Cuando este servidor recibe el paquete encapsulado, lo “desencapsula” y procesa la petición. Finalmente devuelve el resultado directamente al usuario de acuerdo a su propia tabla de enrutamiento.

Cuando la conexión finaliza o muere (timeout), el registro de la conexión será eliminado de la tabla “hash”.

La siguiente imagen muestra el flujo de los datos anteriormente descrito.



Notar que todos los servidores pueden tener cualquier dirección IP real de cualquier red incluso pueden estar geográficamente distantes, pero **deben** soportar el protocolo de encapsulación IP. Sus dispositivos de túnel deben ser configurados para que el sistema pueda “desencapsular” correctamente los paquetes recibidos (encapsulados), y la IP virtual debe ser configurada con **dispositivos no-arp** o alias de **dispositivos no-arp**, o bien el sistema puede ser configurado para redireccionar paquetes para la IP virtual hacia el socket local.

Finalmente, cuando un paquete encapsulado llega, el servidor real lo desencapsulará y notará que el paquete está destinado a la IP Virtual, entonces dirá “Oh, es para mí”, entonces procesará la petición y devolverá los resultados directamente al usuario final.

Virtual Server vía Ruta Directa.

Al igual que en el Virtual Server vía IP Tunneling, LinuxDirector procesará solo la mitad cliente-servidor de una conexión en el VS vía Ruta Directa, y las respuestas pueden seguir rutas separadas hacia los clientes. Esto puede incrementar la escalabilidad del Virtual Server de forma considerable.

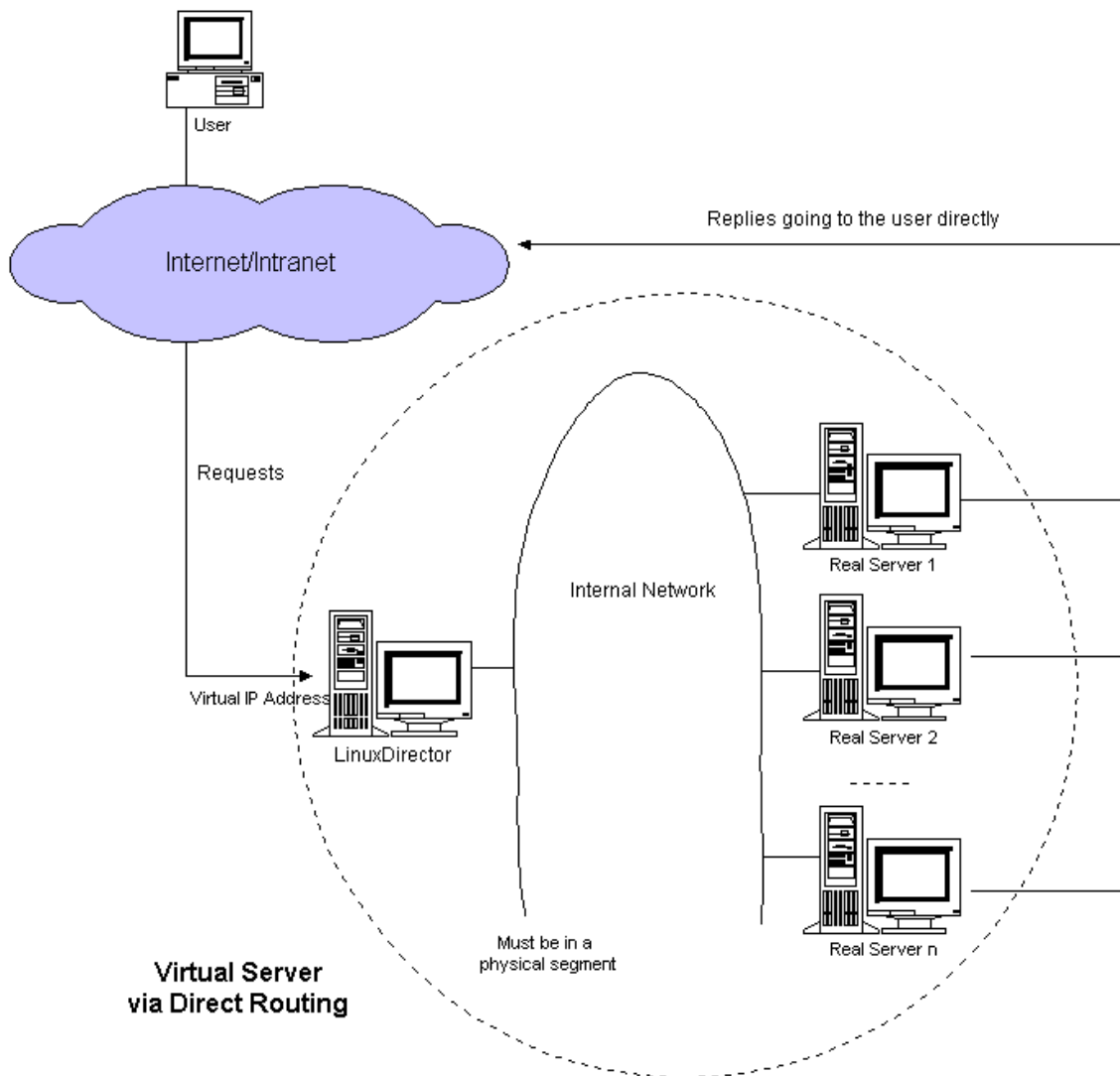
Comparado con el VS vía IP Tunneling, este diseño no posee túnel “overhead”, pero requiere que una de las interfaces de los balanceadores de carga y una de las interfaces del servidor real estén en el mismo segmento físico.

Funcionamiento.

El acercamiento a las peticiones de despacho es similar al utilizado en el NetDispatcher de IBM. La dirección IP virtual es compartida por los servidores reales y el de balanceo de carga. Este último también debe tener una interfaz configurada con la IP virtual, la que será usada para aceptar la petición de paquetes y enrutarlos directamente al servidor elegido.

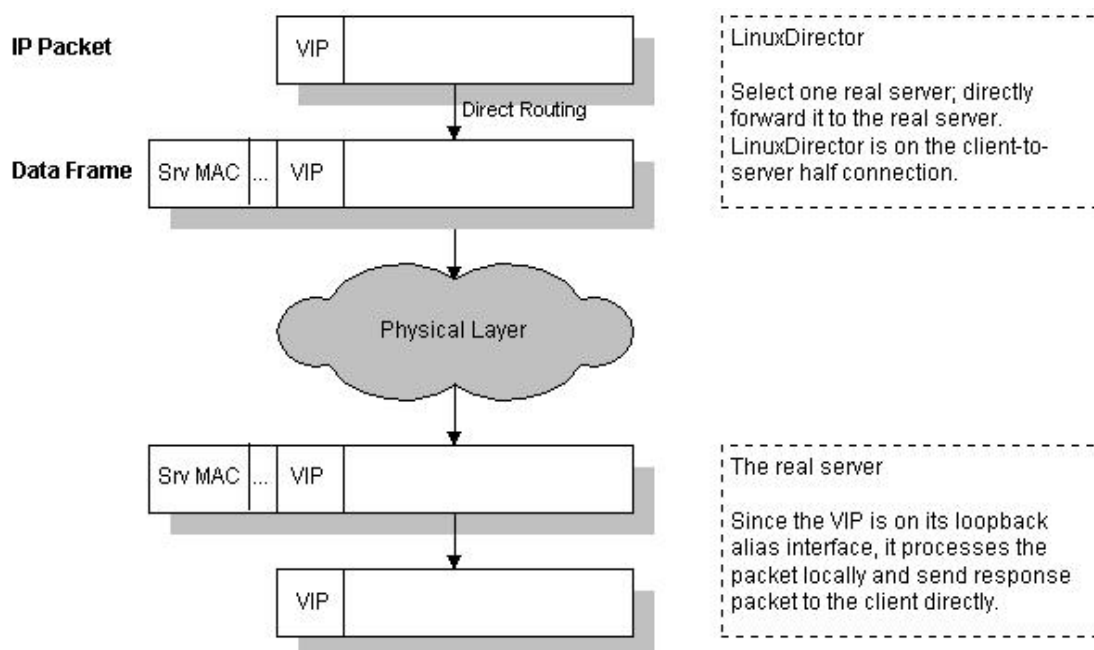
Todos los servidores reales deben tener la interfaz **no-arp** configurada con la dirección IP virtual o bien redireccionar los paquetes destinados para la dirección IP virtual a un socket local, para que así el servidor real pueda procesar los paquetes localmente.

El balanceador de carga y los servidores reales deben tener una de sus interfaces físicamente conectadas por un hub o switch:



Cuando un usuario accede a un servicio virtual, proporcionado por el servidor Cluster, llegará un paquete destinado a una IP virtual. El balanceador de carga (LinuxDirector) examinará la dirección y puerto de destino del paquete. Si ambos coinciden para un servicio virtual, un servidor real es elegido por el algoritmo de programación, y la conexión se agrega en la tabla hash con las conexiones grabadas. Entonces el balanceador de carga reenvía (forward) el paquete al servidor elegido. Cuando el paquete entrante perteneciente a esta conexión y al servidor elegido aparece en la tabla hash, el paquete será nuevamente enrutado directamente al servidor. Cuando el servidor recibe el paquete, verá que es para la dirección en su propia interfaz o bien en un socket local, entonces procesará la petición y la respuesta directamente con el usuario final. Luego de que la conexión termina o muere, se eliminará de la tabla hash.

La siguiente imagen muestra el flujo de los datos anteriormente descrito.



Autor.
 Boris Quiroz Q.
<http://boris.penguin.cl>
 Versión 1.0
 cc by-nc-nd/2.0/cl

