Presentació per a l'assignatura ASO-FIB

Sistemes web d'alta disponibilitat

Pau Freixes pfreixes@milnou.net

2006-12-11

Índex

Model clàssic per desplegar un servidor web

El model de granja "The farm"

Einees en GNU/Linux per aplicar el model de granja

Problemes implícits en el protocol web i arquitectura

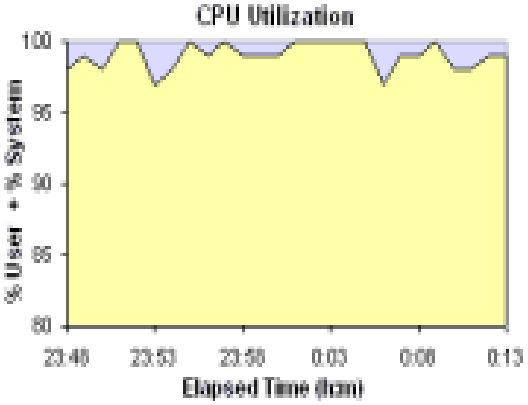
Configuració del failover i distribució de carga

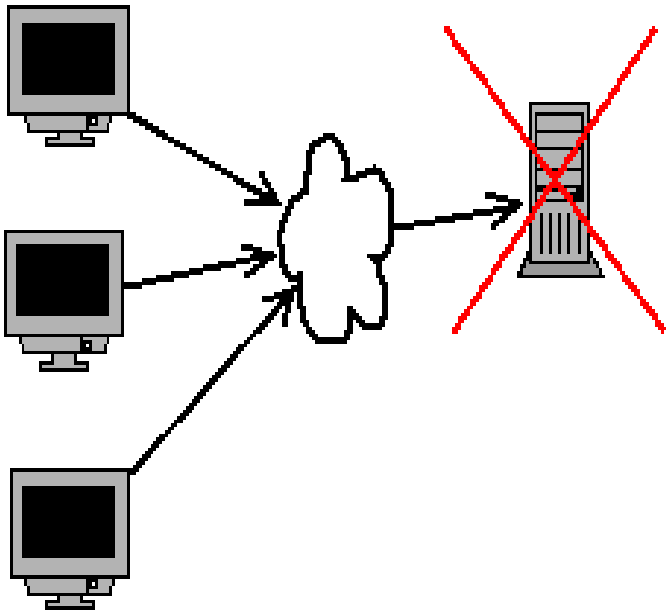
Model clàssic per desplegar un servidor web

Desplegar un servidor web és una feina “senzilla”, però l'arquitectura bàsica sobre el que es desplega no és suficient per tenir un servei 24x7

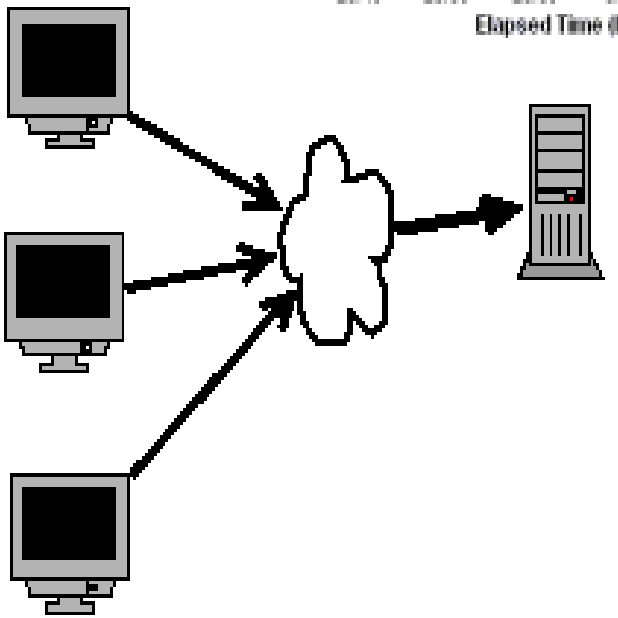
Model clàssic per desplegar un servidor web

La problemàtica del failover i de la distribució de carga en un entorn bàsic, **com el gestionem** ?





failover



distribució de carga

El model de granja "The farm"

Què és millor si augmenta la demanda de càlcul dels nostres sistemes, **augmentar els recursos del nostre servidor** o bé **afegir un nou servidor i repartir equitativament les tasques entre aquests** ? “The farm” és una arquitectura que aposta per la segona opció

El model de granja "The farm"

Què és ?

• És una col·lecció de servidors que es veuen com un gran servidor
• Tots ells executen una mateixa tasca
• La suma de les tasques es veuen com una sola tasca
• La caiguda d'un node només afectarà al rendiment global però no al servei

Usos habituals

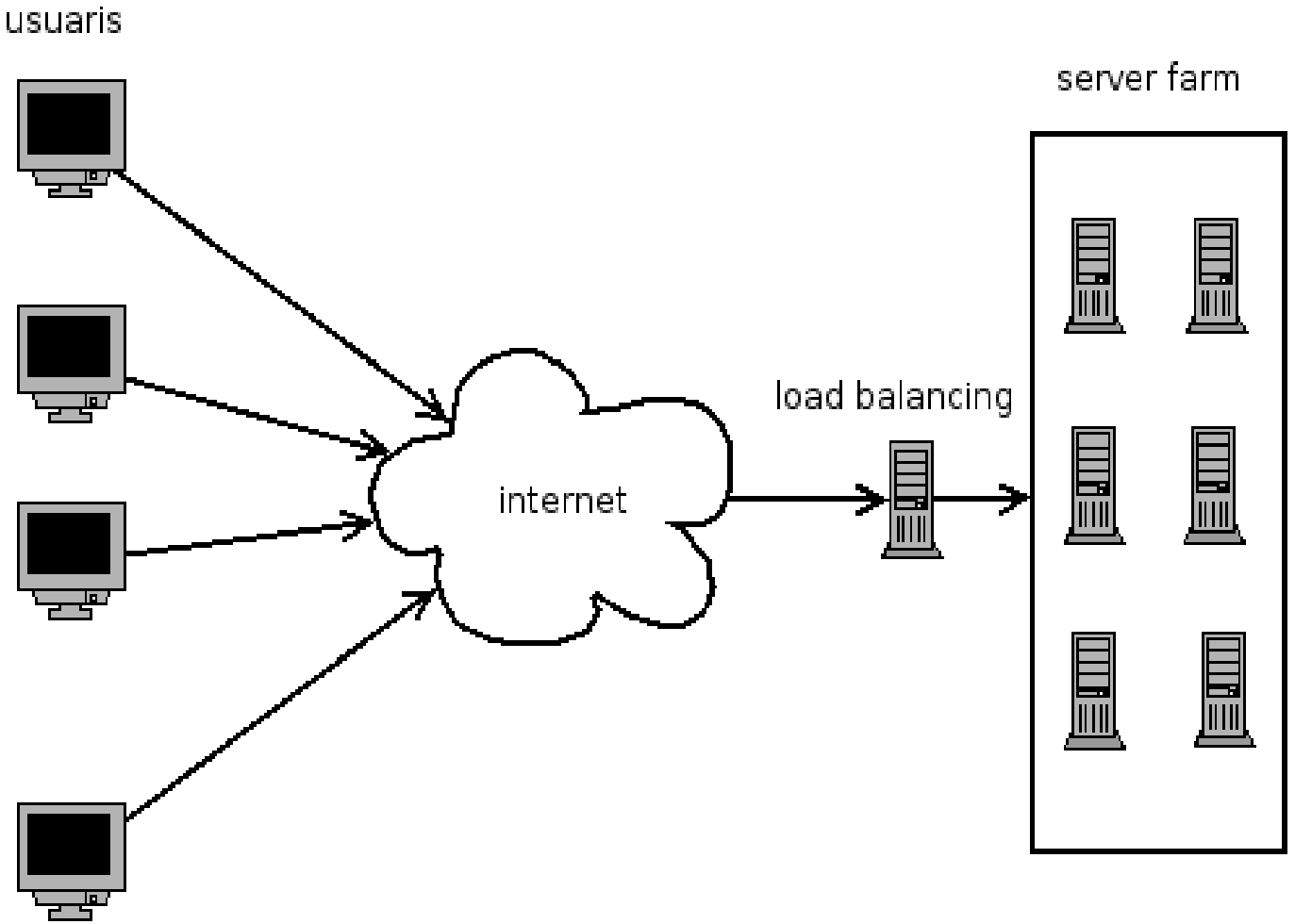
· Renderització de gràfics (*veure blende*r)
· Càlculs matemàtics (centres científics)
· Serveis d'alta disponibilitat web, dns, etc (*google, youtube, akamai* ...)

Independència de les tasques dels nodes

Cal diferenciar com a mínim dos tipus de models, aquell on **les tasques que s'executen en els nodes són independents a les tasques dels altres nodes** i un altre on tasques depenen d'altres tasques que s'executen en altres nodes, **nosaltres ens concentrarem en el primer cas**, menys complexitat i és d'ús habitual per a servis com el web, dns o altres

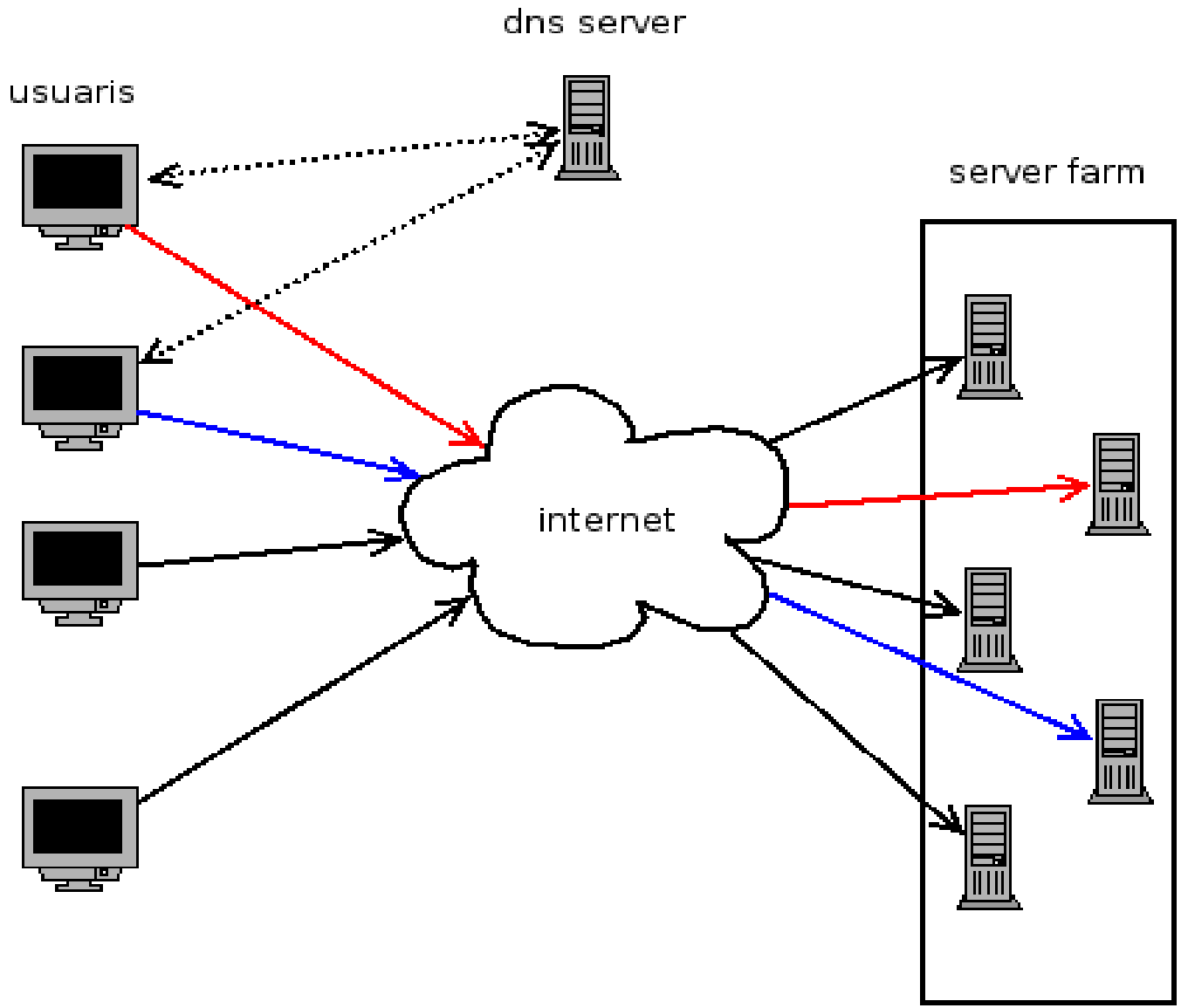
El model de granja "The farm"

Tres models diferents, local balancing



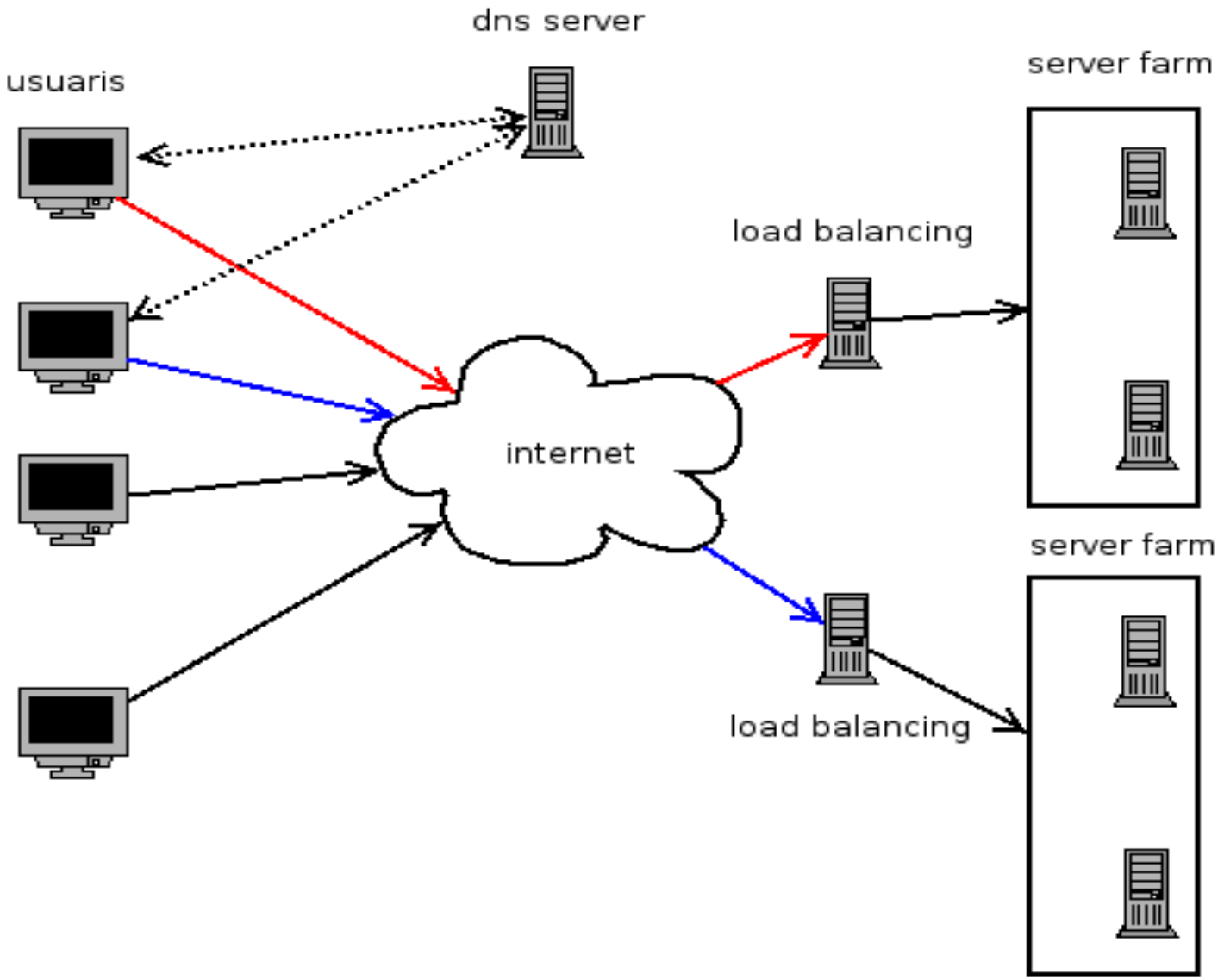
El model de granja "The farm"

Tres models diferents, dns balancing



El model de granja "The farm"

Tres models diferents, mixed : local balancing + dns balancing



Dns versus Local balancing

On hi ha el problema de càrrega ?

- A l'ample de banda, DNS BALANCING
- A les aplicacions, LOCAL BALANCING o DNS BALANCING
- A tots dos llocs , DNS BALANCING o (DNS BALANCING i LOCAL BALANCING)

Cal comprendre primer de tot on s'està quedant petit el sistema, no serveix de res escalar les aplicacions si el problema és a l'ample de banda !!

Altres consideracions:

- Si cal distribuir de forma geogràfica per reduir el temps de comunicació és habitual utilitzar un servei de DNS balancing (www.akamai.com)
- Quin és el pressupost ? **baix => LOCAL BALANCING**

El model de granja "The farm"

Si hem d'aplicar un model de granja amb **Local Balancing** cal tenir algunes coses en ment

Avantatges

· Permet escalar de forma eficient la demanda als diferents nodes
· Permet aplicar polítiques de failover
· Ampliar la capacitat global de forma senzilla

Inconvenients

· La xarxa es pot convertir en un punt crític
· El distribuïdor de càrrega és l'element crític
· El número de servidors a mantenir és més elevat
· Sorgeixen un conjunt de problemàtiques implícites a la nova arquitectura
  · Autenticació centralitzada d'usuaris ?
  · Centralització de les dades comunes, un punt crític ?
  · Problemes del protocol a escalar, les sessions en el protocol web ?

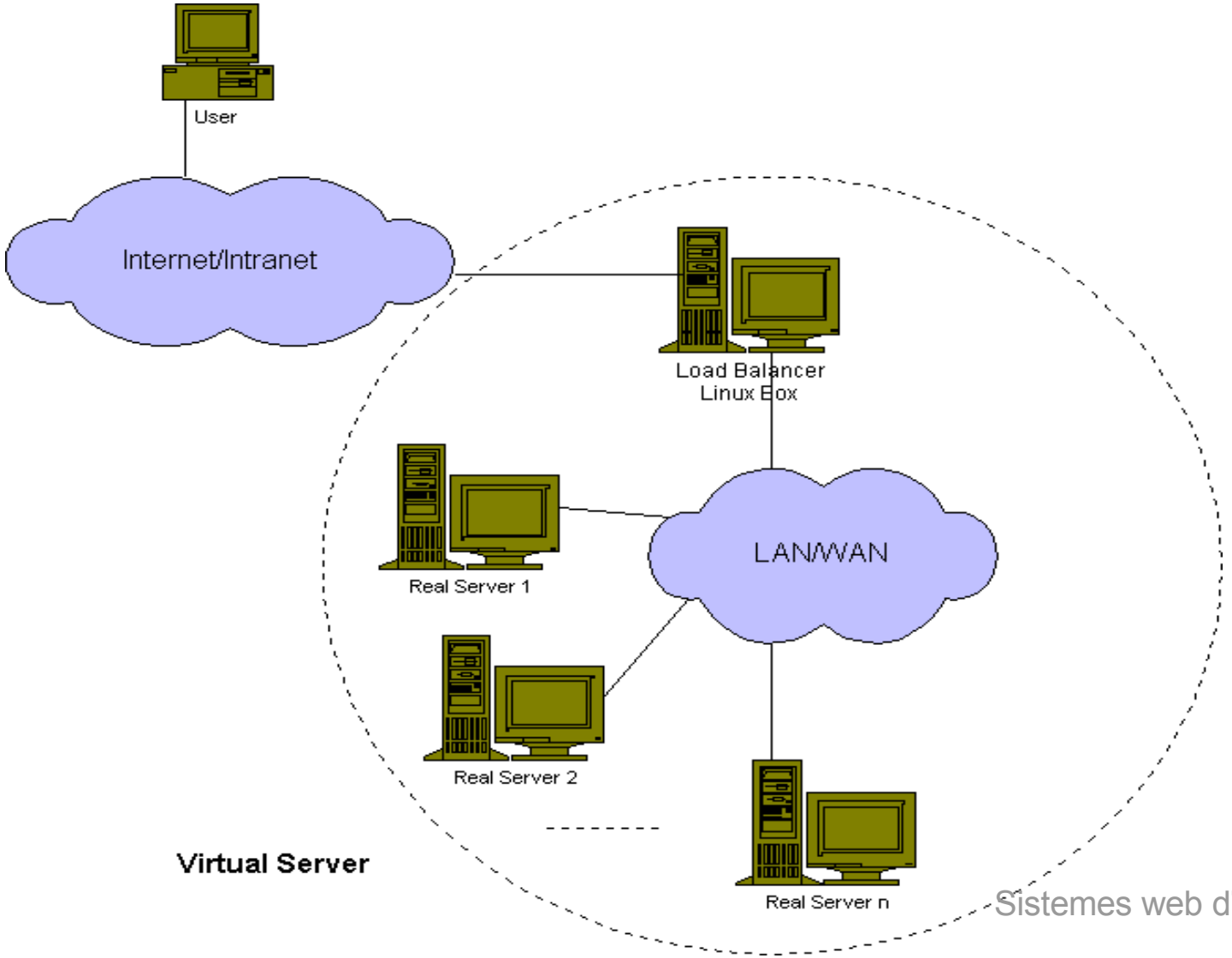
Einees en GNU/Linux per aplicar el model de granja

A l'univers GNU Linux existeixen algunes eines per poder aplicar aquest model de granja, nosaltres ens concentrarem en dues d'elles

- Linux Virtual Server
- Monit

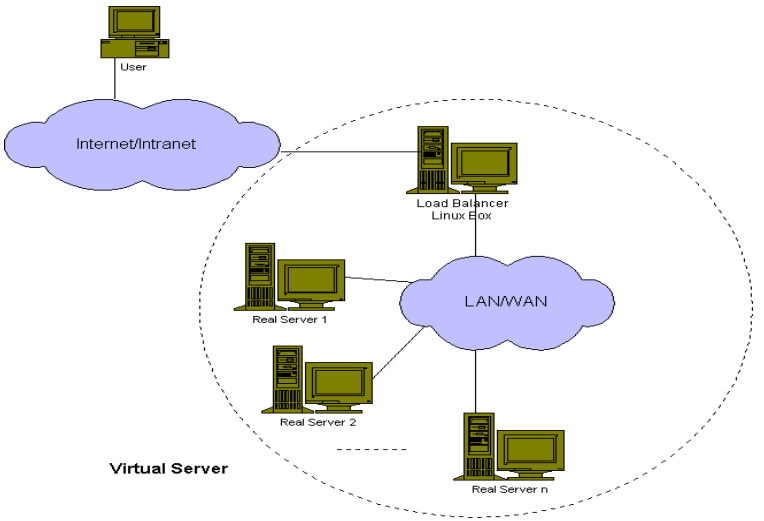
Linux Virtual Server, definició

" Linux Virtual Server és un servidor virtual altament escalable format per un cluster de servidors linux reals "



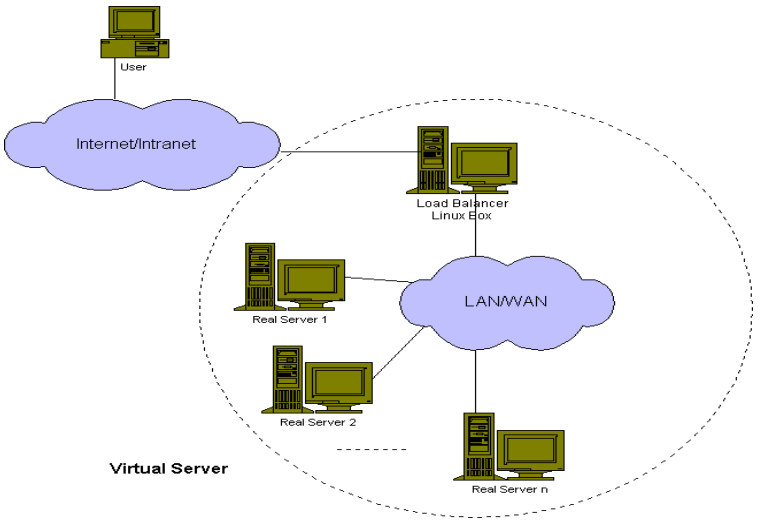
Linux Virtual Server, característiques

- Composat per un modul del kernel i amb un conjunt d'eines en espai d'usuari
- Multituds de “Virtual Servers” en una mateixa configuració
- Polítiques de filtrat
  - tcp
  - udp
  - mark
- Polítiques de distribució (scheduling method)
  - round robin
  - hashing
  - weight
  - shortest path



Linux Virtual Server, característiques

- Polítiques de redirecció (forwarding method)
  - gateway
  - ipip
  - masquerading



Linux Virtual Server, ipvsadm

Ipvsadm és l'eina en espai d'usuari que se'ns subministra per a poder configurar els nostres Virtual servers, l'ordre pot seguir tres patrons amb tres funcionalitats diferents

· Afegir un nou virtual server

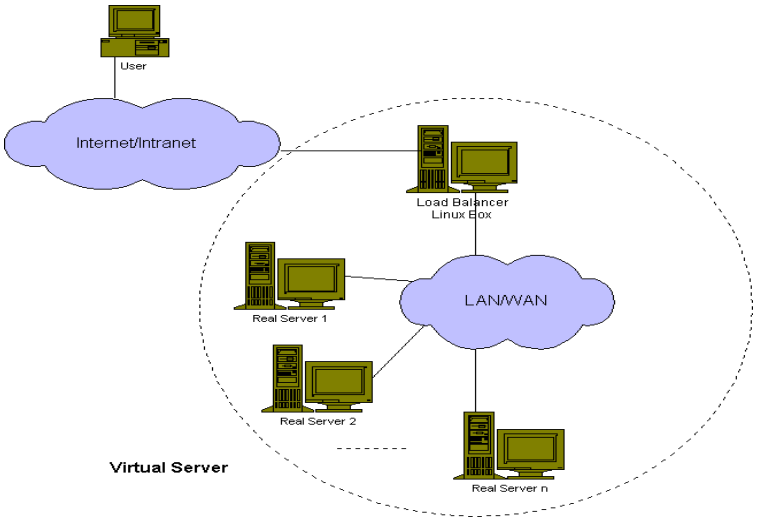
ipvsadm -A <política de filtratge> -s <política de balanceig>

· Afegir servidors físics a un virtual server

ipvsadm -a <política de filtratge> -r <real_server> <política de redirecció>

· Control de l'activitat dels virtual servers

ipvsadm -l



Linux Virtual Server, compatibilitat amb iptables

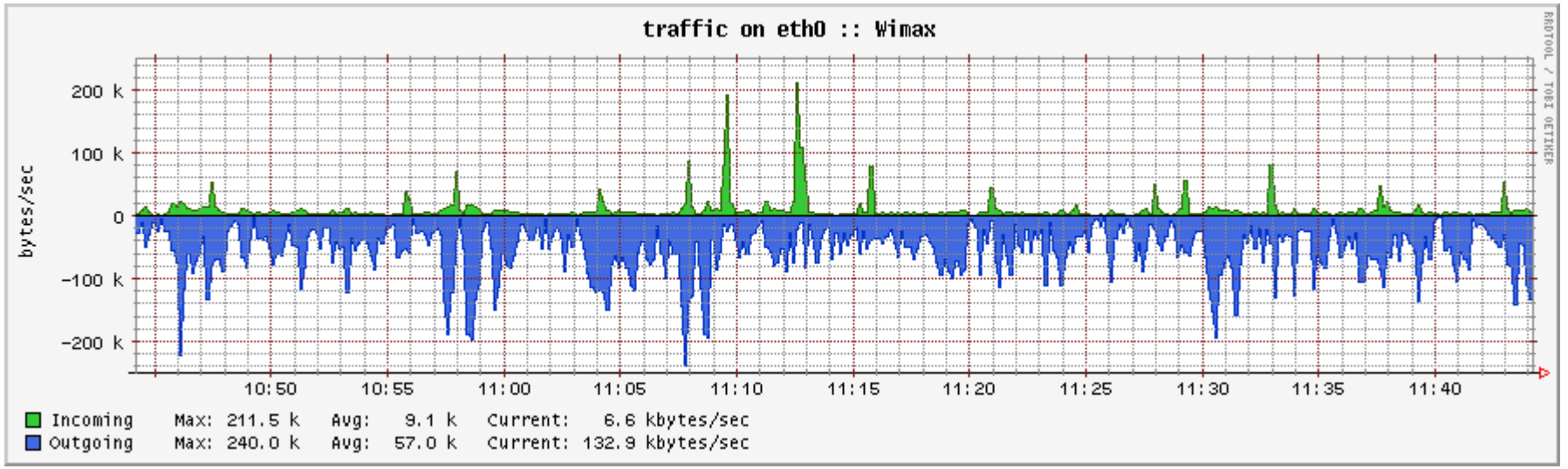
És força habitual fer accounting del número de bytes mitjançant **iptables** del tràfic local

```
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 80 -j ACCEPT
```

↓

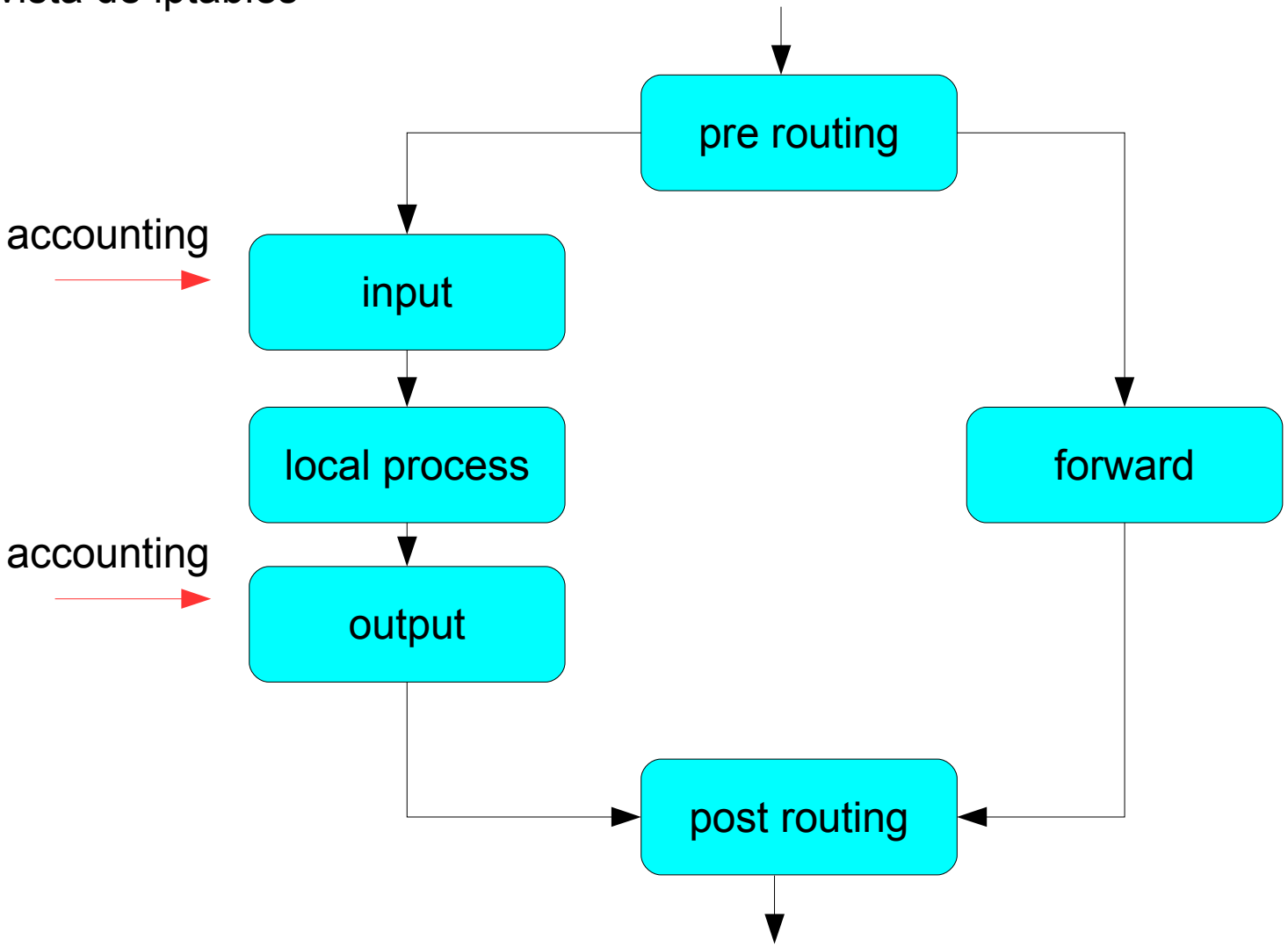
PYTHON | PERL | PHP + RRD

↓

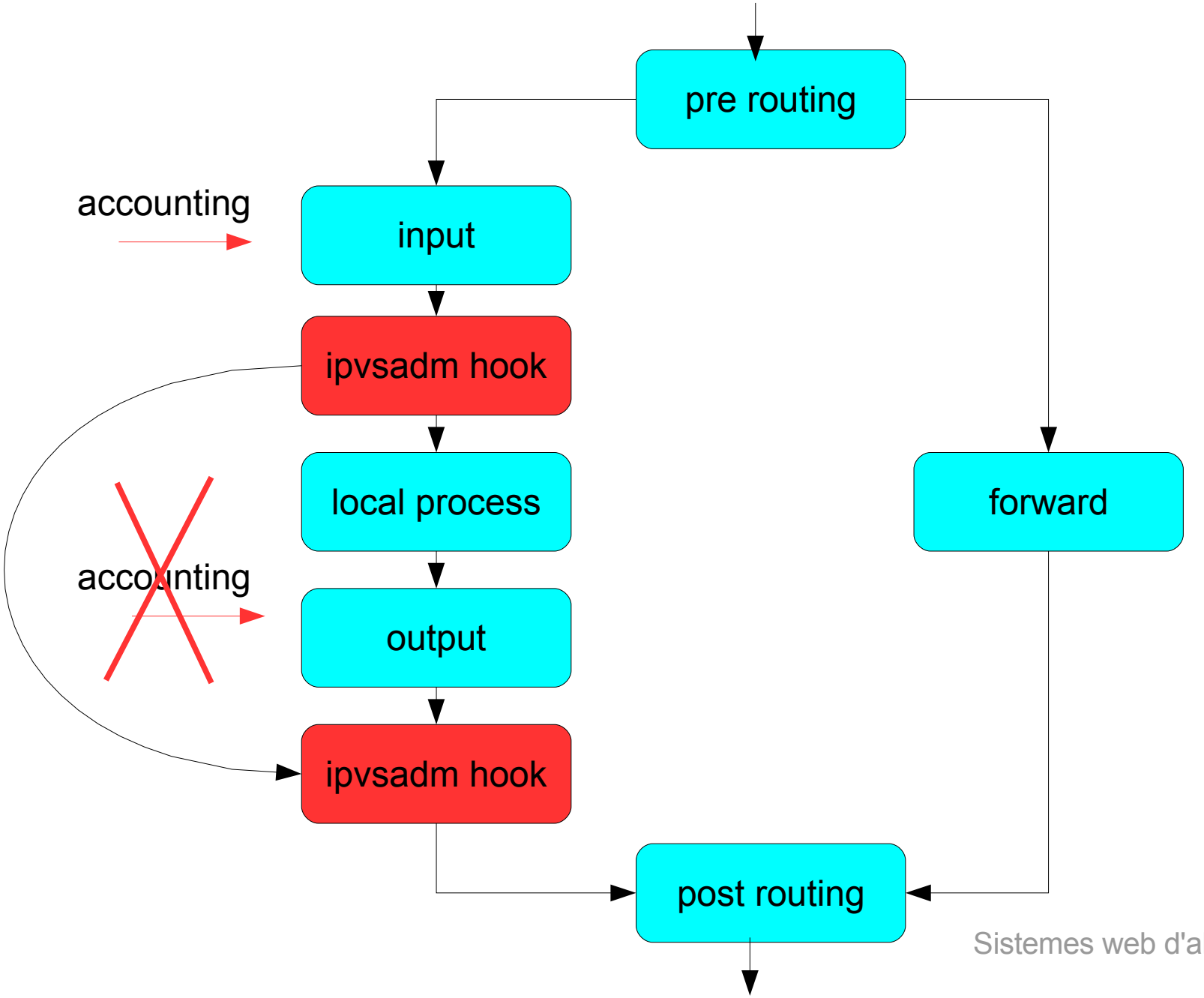


Linux Virtual Server, compatibilitat amb iptables

Gràfic **minimitzat** dels passos que segueix un paquet en el kernel des del punt de vista de iptables

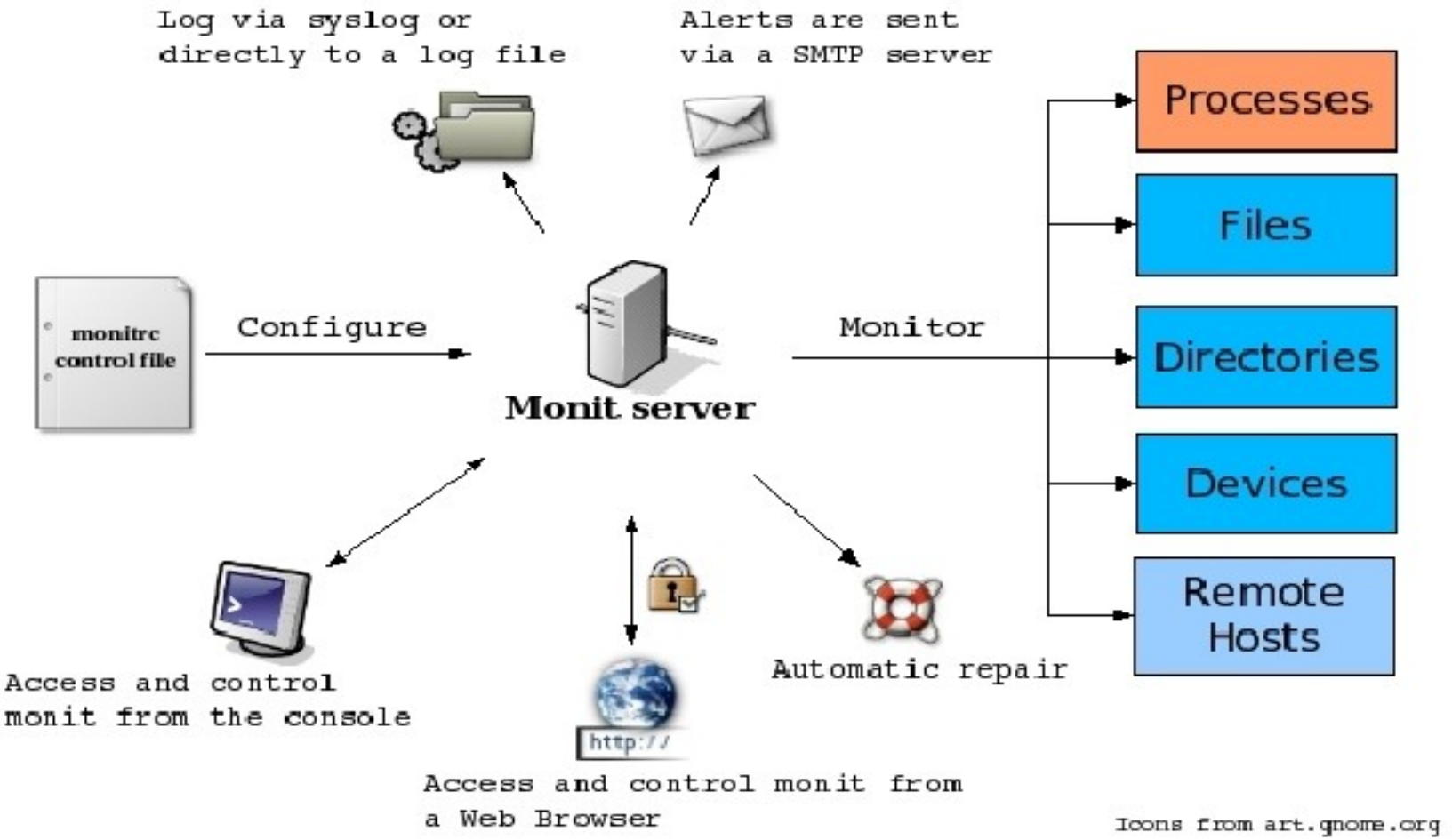


Linux Virtual Server, compatibilitat amb iptables



Monit, definició

" Utilitat per monitoritzar arxius, directori, devices, **serveis i hosts** en un entorn unix "



Monit, característiques

· S'executa en background en el sistema com a dimoni

· Arxiu de configuració complex (/etc/monit/monitrc)

· Tipus de tests o checks

  · per arxius (normals, directoris, devices)

  · per procés

  · per servei (tcp, udp)

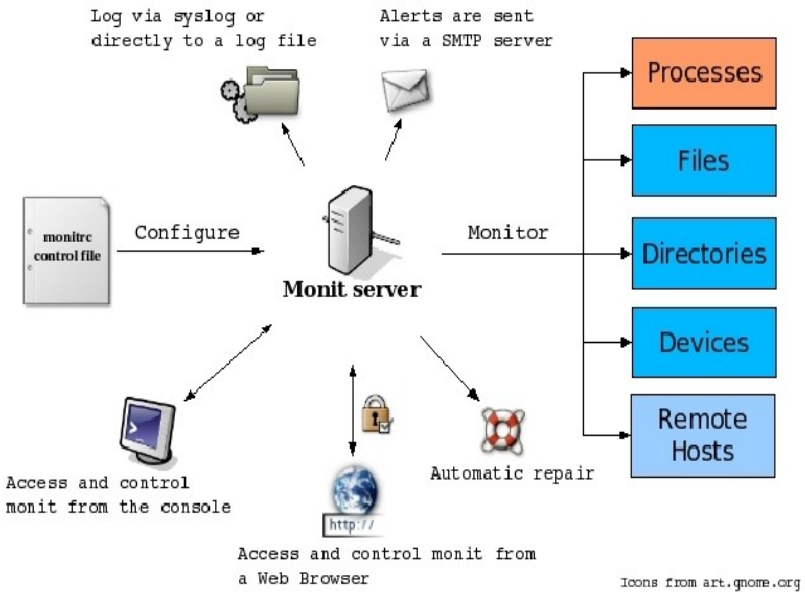
  · per host (icmp)

  · per sistema (cpu, espai en disc, ...)

· Sistema d'events

  · email

  · execució d'un script



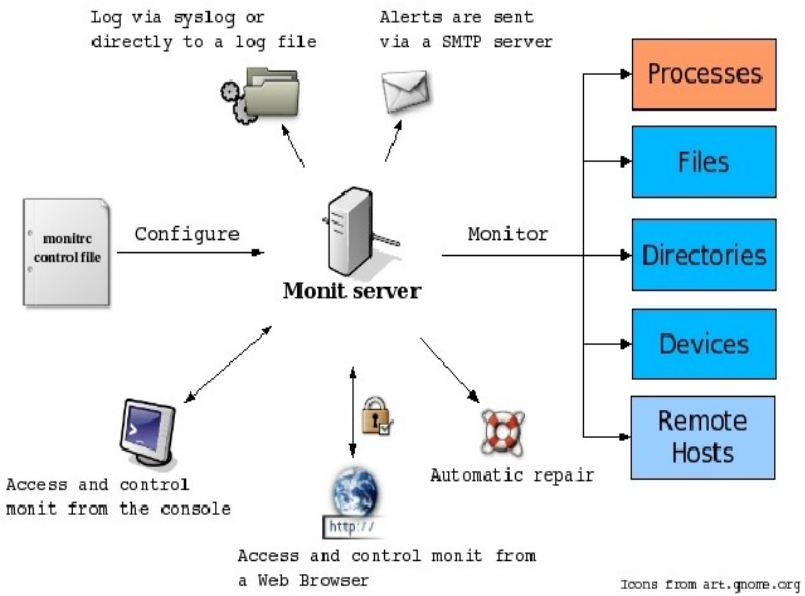
Monit, configuració

Patró de configuració de monitrc per a cheks de serveis – web, dns, smtp ... - de màquines del virtual server, arxiu **/etc/monit/monitrc**

check host *nom_del_host* with address *ip_del_host*
start program =”*./script_executar.sh*”
if failed port *servei_a_seguir* (tcp|udp) then start
if *n_vegades* restarts within *n_cicles* then timeout

Exemple de com configurar monit per seguir el servei de dns local

check host firewall with address 127.0.0.1
start program =”/etc/init.d/named start”
if failed port 53 udp then start
if *1* restarts within *2* then timeout



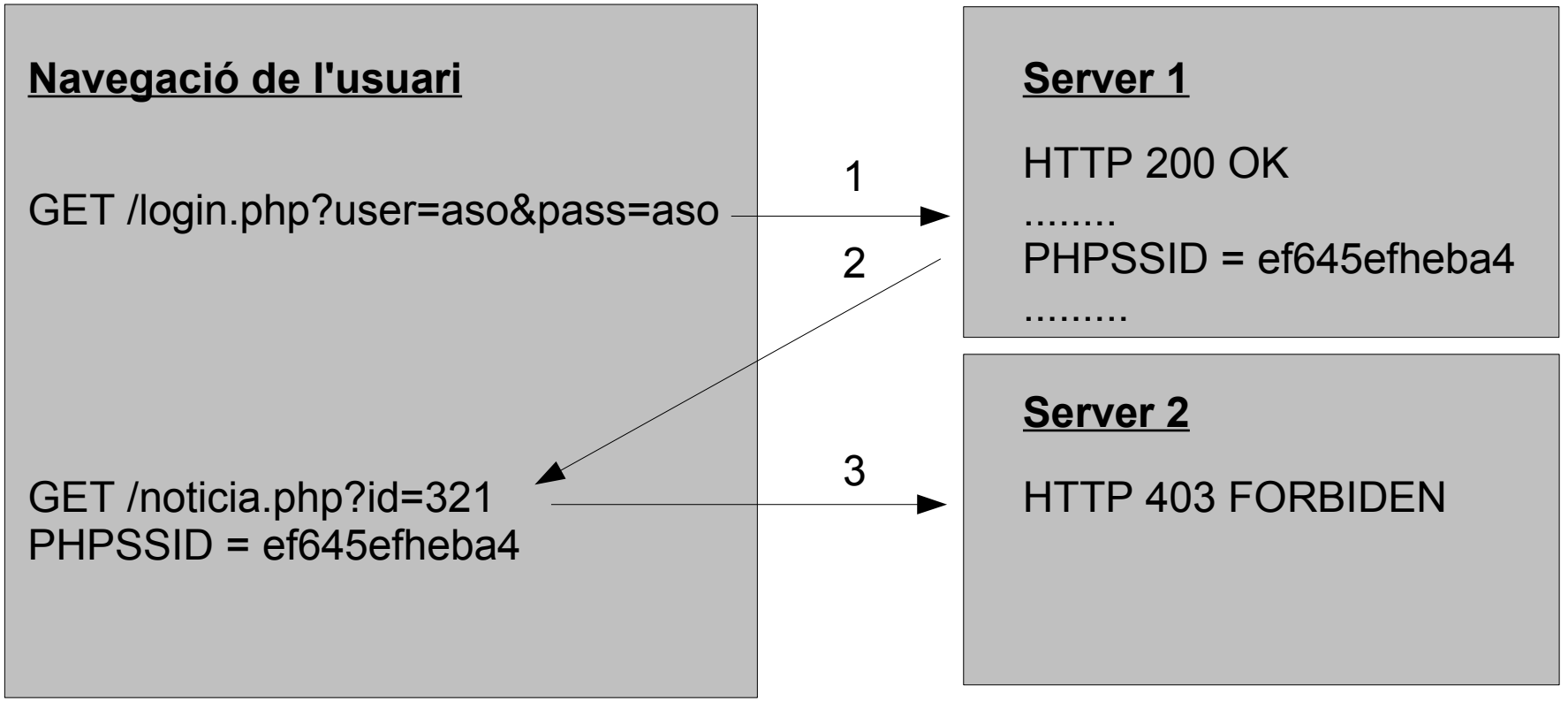
Problemes implícits en el protocol web i arquitectura

Modificar l'arquitectura i introduir canvis pot portar problemes
En el cas que ens ocupa analitzarem dos dels problemes que caldria tenir en compte i solucionar.

- El problema de les sessions
- La descentralització de la informació

El Problema de les sessions

· El protocol web és un protocol sense estat per definició, però l'ús de sessions modifica aquesta afirmació. La informació associada a una sessió s'emmagatzema al servidor web, que passarà si la navegació d'un usuari salta entre els diferents servidors de la granja ?

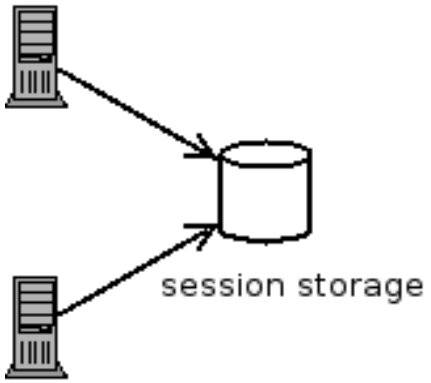


El Problema de les sessions

· Evitar que una mateixa connexió d'un usuari pugui saltar de servidors. **Ipvsadm** permet aplicar l'algoritme de balanceig anomenat Source Hashing .

$$f = ip \bmod n\_serv$$

· Implementar un sistema de sessions amb emmagatzematge en base de dades.



Problemes implicits en el protocol i arquitectura

· La descentralització de la informació

·Un servidor web necessita d'una configuració i de les aplicacions – html, php, etc - per poder funcionar, en un entorn “the farm” podem optar per dos models diferenciats

Cada servidor té les seves dades

· Pocs servidors, es pot fer a ma però condueix a errors humans
· Molts servidors, cal aplicar sistemes automàtics com ara rsync
· No es recomanable en un entorn com ara proveïdors de hosting

Tots els servidors comparteixen un punt d'emmagatzematge a la xarxa

. Els canvis són immediats a tots els servidors
. Tornem a tenir un punt crític

Configuració del failover i distribució de carga

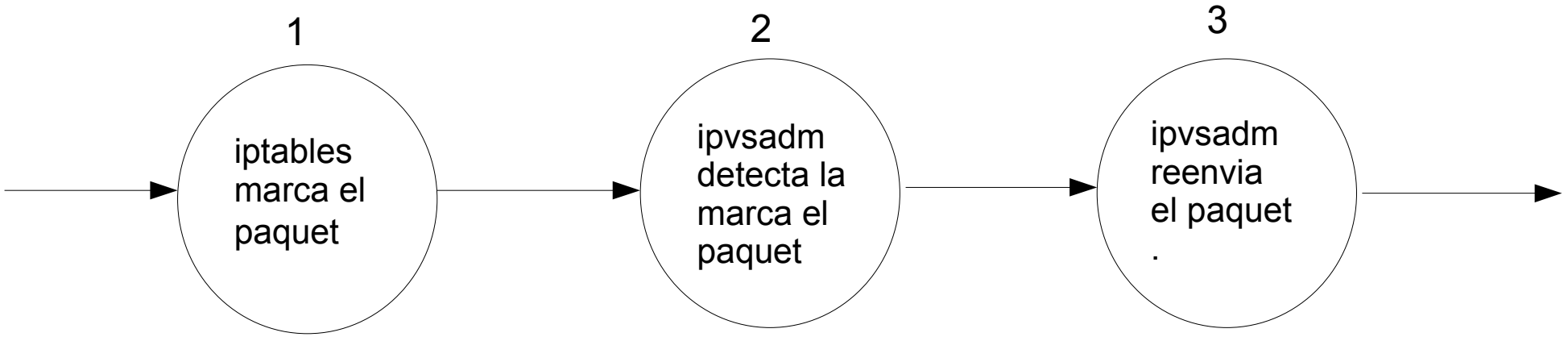
Exemple minimalista de configuració d'una granja de webservers utilitzant les eines esmentades i superant els problemes derivats de l'ús del protocol web

ipvsam

· Política de scheduling Source Hasing, d'aquesta forma totes les connexions realitzades per un usuari seran servides pel mateix servidor, solucionem el problema de les sessions *(2)*

· Model de redirecció de paquets anomenat “Masquerading”, els servidor web rebran com ip d'origen la del servidor on s'està executant ipvsadm *(3)*

· Utilitzem marques per identificar diferents configuracions de ipvsadm, ens serà util per a donar d'alta i baixa servidors en temps real i fer un binding amb monit pel tractament del failover. Per poder utilitzar el sistema de marques haurem de treballar amb iptables *(1)*

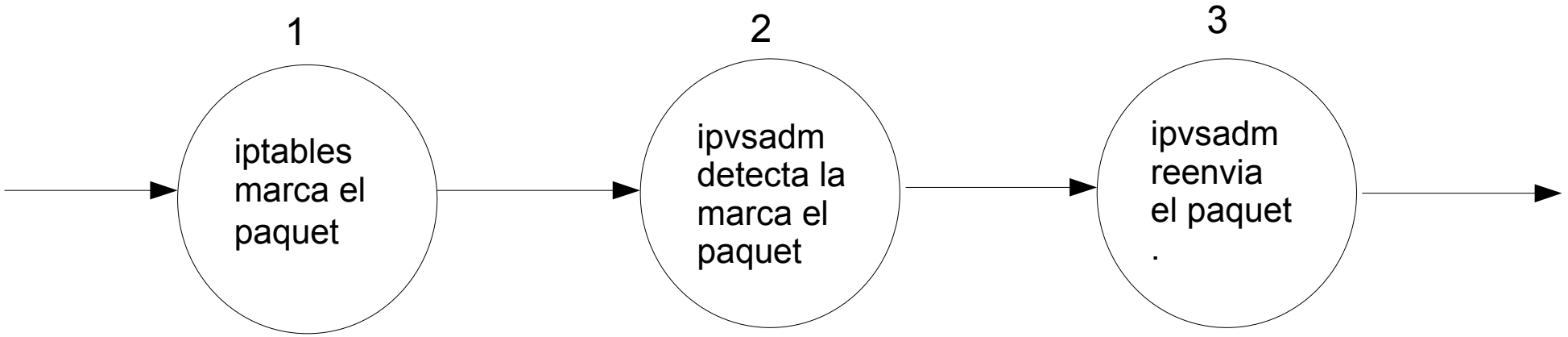


Passos – minimitzats – que fa el paquet en el moment de travessar el kernel

Configuració del failover i distribució de carga

```
iptables  -A PREROUTING -t mangle -d $IP_WEB -p tcp --dport 80 -j MARK --set-mark 1

ipvsadm -A -f 1 -s sh
ipvsadm -a -f 1 -r $IP_SERVER1:80 -m
ipvsadm -a -f 1 -r $IP_SERVER2:80 -m
```



Passos – minimitzats – que fa el paquet en el moment de travessar el kernel

Configuració del failover i distribució de carga

monit

· Farem tantes entrades a l'arxiu de configuració com servidors web a la granja tinguem

· Per a cada entrada monit farà un seguiment del port del servei web per saber si el node a que fa referència l'entrada es troba actiu o no

· En cas de detecció de caiguda executarem un script

```
check host web_server_1 with address $IP_SERVER1
        start program = "web_balancing.sh dead 1"
        if failed port 80 then start
        if 1 restarts within 2 cycles then timeout
```

Entrada a l'arxiu /etc/monit/monitrc

Configuració del failover i distribució de carga

ipvsam + monit + 2 servers cas practic

```
start)
iptables  -A PREROUTING -t mangle -d $IP_WEB -p tcp --dport 80 -j MARK --set-mark 1

ipvsadm -A -f 1 -s sh
ipvsadm -a -f 1 -r $IP_SERVER1:80 -m
ipvsadm -a -f 1 -r $IP_SERVER2:80 -m

ipvsadm -A -f 2 -s sh
ipvsadm -a -f 2 -r $IP_SERVER1:80 -m

ipvsadm -A -f 3 -s sh
ipvsadm -a -f 3 -r $IP_SERVER2:80 -m
;;
dead)

iptables -F PREROUTING -t mangle
if [ $2 = 1 ]; then
        iptables  -A PREROUTING -t mangle -d $IP_WEB -p tcp --dport 80 -j MARK --set-mark 3
else
        iptables  -A PREROUTING -t mangle -d $IP_WEB -p tcp --dport 80 -j MARK --set-mark 2
fi
```

Part del script web_balancing.sh

Configuració del failover i distribució de carga

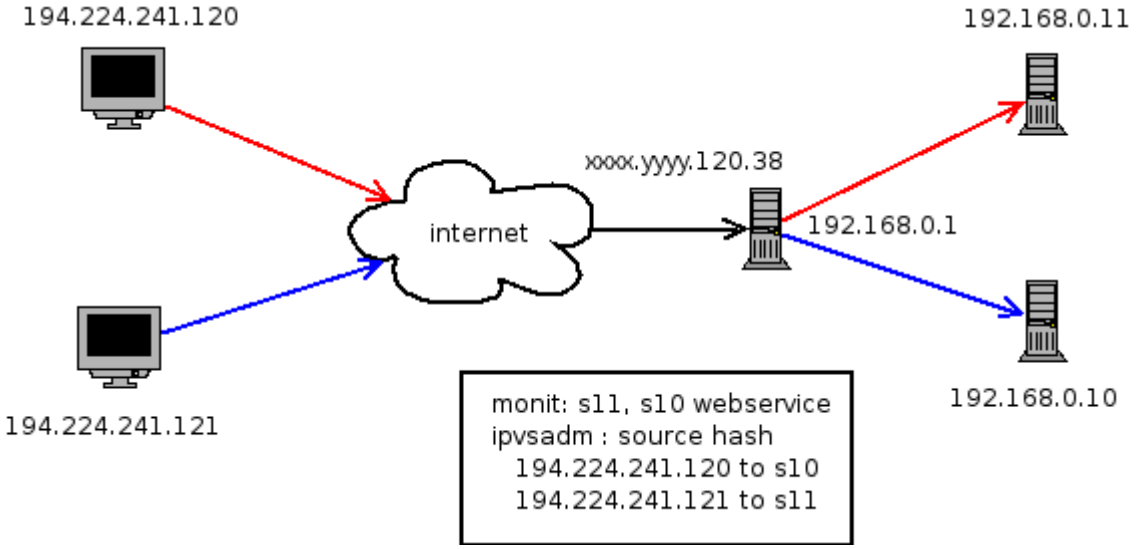
ipvsam + monit + 2 servers cas pràctic

```
check host web_server_a with address $IP_SERVER1
        start program = "/web_balancing.sh dead 1"
        if failed port 80 then start
        if 1 restarts within 2 cycles then timeout

check host web_server_b with address $IP_SERVER2
        start program = "web_balancing.sh dead 2"
        if failed port 80 then start
        if 1 restarts within 2 cycles then timeout
```

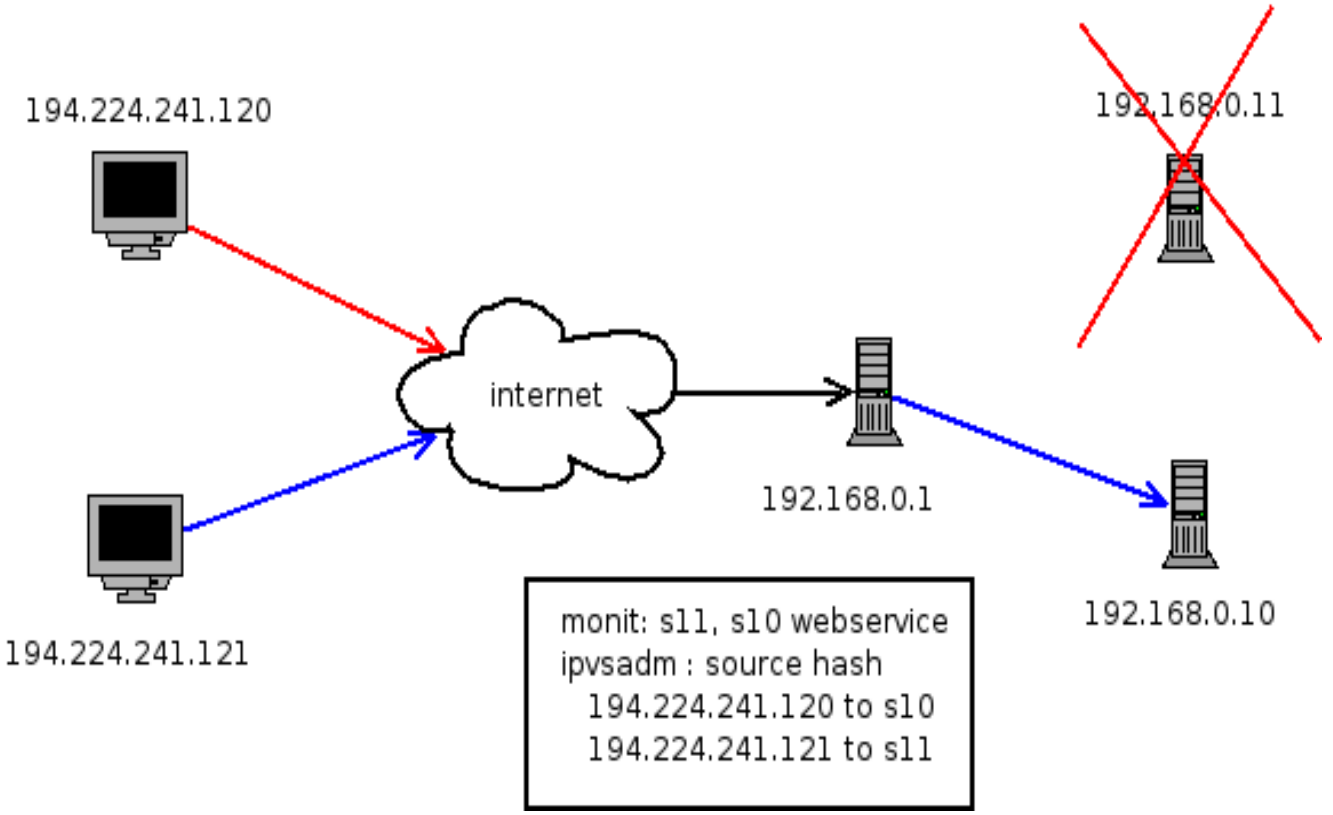
Entrades a l'arxiu de configuració del monit

Configuració del failover i distribució de carga

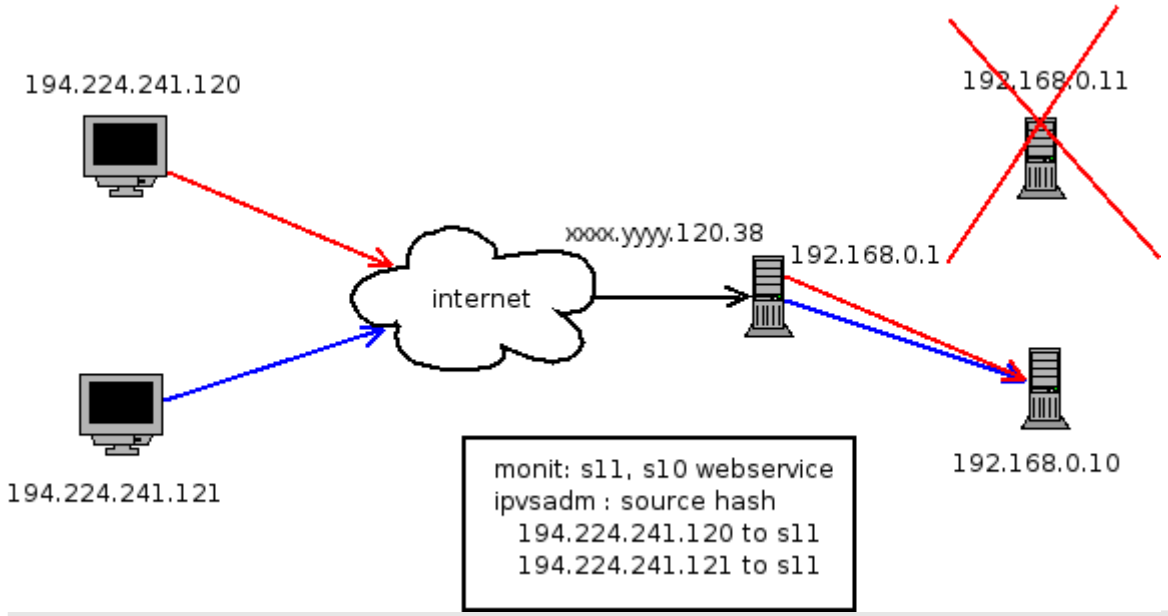



```
firewall9LMDS:~ # pvsadm -l
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
FWM  1 sh
  -> s11.mynews-intranet:http     Masq    1      5          6
  -> s10.mynews-intranet:http     Masq    1      8          4
FWM  2 sh
  -> s10.mynews-intranet:http     Masq    1      0          0
FWM  3 sh
  -> s11.mynews-intranet:http     Masq    1      0          0
firewall9LMDS:~ # iptables -nvL -t mangle
Chain PREROUTING (policy ACCEPT 687M packets, 492G bytes)
 pkts bytes target     prot opt in     out     source               destination
1134K  119M MARK       tcp  --  *      *       0.0.0.0/0            xxx.yyyy.120.38      tcp dpt:80 MARK set 0x1
```

Configuració del failover i distribució de carga



Configuració del failover i distribució de carga




```
firewall9LMDS:~ # pvsadm -l
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
FWM  1 sh
  -> s11.mynews-intranet:http     Masq    1      0          6
  -> s10.mynews-intranet:http     Masq    1      0          4
FWM  2 sh
  -> s10.mynews-intranet:http     Masq    1      4          2
FWM  3 sh
  -> s11.mynews-intranet:http     Masq    1      0          0
firewall9LMDS:~ # iptables -nvL -t mangle
Chain PREROUTING (policy ACCEPT 687M packets, 492G bytes)
 pkts bytes target     prot opt in     out     source               destination
1134K  119M MARK       tcp  --  *      *       0.0.0.0/0            xxx.yyyy.120.38      tcp dpt:80 MARK set 0x2
```

http://en.wikipedia.org/wiki/Server_farm

http://www.linuxvirtualserver.org/

http://www.tildeslash.com/monit/