Linux Esotico – 2
il ritorno del pinguino mannaro

Alberto Cammozzo
mmzz@stat.unipd.it

Mauro Luzi
mluzi@pluto.it

Serate a tema PLUTO Padova
19 Maggio 2004

Linux Esotico: Sommario

- V-server
- User Mode Linux

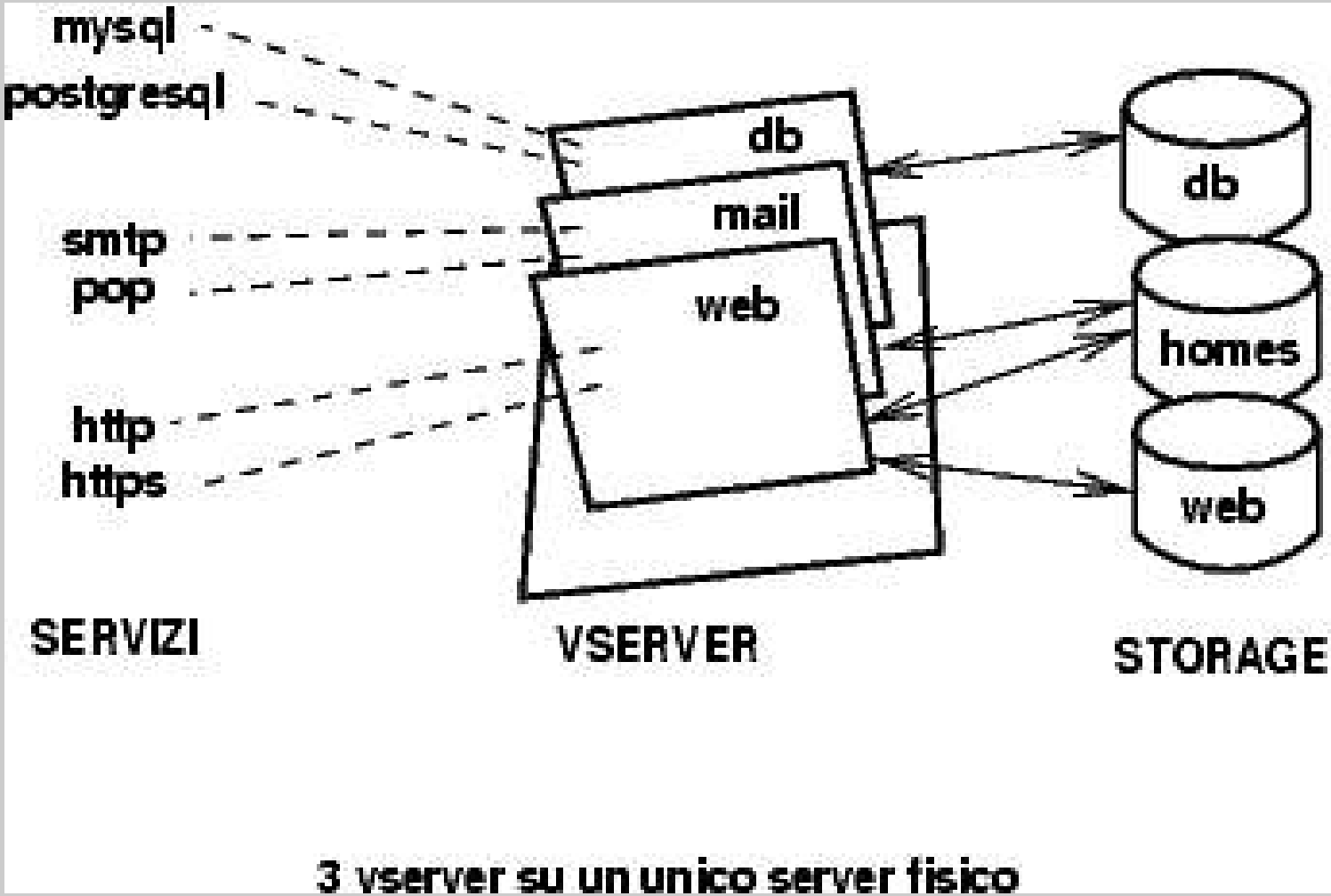
V-server

- Un solo server hardware
- Un solo kernel 2.4.X + patch *ctx*
- Diversi server indipendenti con indipendenza di:
  - spazio processi
  - networking TCP/ IP e Sys V IPC
  - filesystem
  - distribution
  - servizi
  - utenti (incluso root)

Vserver: a cosa serve?

- **Hosting**: basta un server hw per diversi clienti
- **Sperimentazione**: coesistenza diverse release
- **Didattica**: un server per studente
- **Security box**: isolamento servizi
- **Migrazione soffice**: da vecchio a nuovo server senza reboot, wow!
- **Rightsizing**: compro solo i server che mi servono.

Vserver: uno schema



3 vserver su un unico server fisico

Creazione del server

- Applicare patch ctx al kernel, reboot
- Installare le utilities: util-vserver (meglio se con prefix = /)
- Creare un server campione: `debootstrap woody wikkit`
- Configurarlo:

```
vi /etc/vservers/wikkit.conf
cp /etc/apt/sources.list /vservers/wikkit/etc/apt/
cp /etc/network/interfaces /vservers/wikkit/etc/network/
cp /etc/resolv.conf /vservers/wikkit/etc/resolv.conf
vi /vservers/wikkit/etc/network/interfaces
```

- in `/dev` del vserver lasciare solo i device: `full initctl log null ptmx pts random reboot shm tty urandom xconsole zero`
- Farlo partire ed entrarci:

```
vserver wikkit start
vserver wikkit enter
```

Ulteriori configurazioni

- SSHd richiede configurazione: /etc/sshd/sshd.conf

```
#ListenAddress 0.0.0.0
ListenAddress IP_DEL_SERVER
X11Forwarding yes
X11DisplayOffset 10
X11UseLocalhost no
```

- In generale: fare attenzione che nessun programma faccia il bind di 0.0.0.0, ma stia ad ascoltare solo sul suo indirizzo IP: usare `netstat -l` per capire che succede.
- Problematici: bind, nfs, arp, ping, traceroute.

Manutenzione ordinaria vserver

- aggiornamento simultaneo di tutti i vserver

```
cd /vservers
for i in *
do
    vserver $i exec apt-get update
    vserver $i exec apt-get upgrade
    echo vserver $i done
done
```

- Migrazione del vserver X dal server A al server B

```
ssh A 'vserver X stop'
ssh A 'cd / ; tar pcvf - /vservers/X ' | ssh B 'cd / ; tar pxvf - '
ssh A 'cat /etc/vservers/X.conf' | ssh B 'cat >> /
  etc/vservers/X.conf'
ssh B 'vserver X start'
```

Esempio

```
mother :~# uname -a
Linux mother  2.4.21-ctx17 #5 SMP Fri Aug 22 08:53:25 CEST 2003 i686 unknown
mother :~# vserver-stat
CTX#   PROC QTY       VSZ        RSS  userTIME   sysTIME    UPTIME NAME
0            37      41MB        2kB   1h57m01  49m05.79  33d02h06 root server
2            10     428MB        4kB   1h10m37   4m51.83  32d08h07 zarquon
5            16     457MB        2kB  34m06.95   1m30.06  32d08h05 db
10           15     853MB        7kB  24m03.05  10m43.46  31d08h35 wonko
11           15     200MB        1kB    m55.50   1m30.75  30d04h32 agda
mother :~# vserver zarquon enter
ipv4root is now 147.162.35.4
New security context is 2
root@zarquon:/# ps ax
  PID TTY      STAT   TIME COMMAND
    1 ?        S      1:21 init
 1886 ?        S      0:03 /sbin/syslogd
 1898 ?        S      0:00 /usr/sbin/inetd
 1912 ?        S      0:02 /usr/sbin/cron
 1917 ?        S      0:45 /usr/sbin/apache-ssl
21386 ?        S      0:00 /usr/lib/apache-ssl/gcache 33 /var/run/gcache_port
21441 ?        S      0:00 /usr/sbin/apache-ssl
21442 ?        S      0:00 /usr/sbin/apache-ssl
21443 ?        S      0:00 /usr/sbin/apache-ssl
21444 ?        S      0:00 /usr/sbin/apache-ssl
21445 ?        S      0:00 /usr/sbin/apache-ssl
 9740 pts/1    S      0:00 /bin/bash -login
 9753 pts/1    R      0:00 ps ax
root@zarquon:/# logout
mother :~#
```

I comandi

```
mother :~# vserver
vserver [ options ] server-name command ...

server-name is a directory in /vservers

The commands are:
 build   : Create a virtual server by copying the packages
           of the root server
 enter   : Enter in the virtual server context and starts a shell
           Same as "vserver name exec /bin/sh"
 exec    : Exec a command in the virtual server context
 suexec  : Exec a command in the virtual server context uid
 service : Control a service inside a vserver
           vserver name service service-name start/stop/restart/status
 start   : Starts the various services in the vserver, runlevel 3
 stop    : Ends all services and kills the remaining processes
 running : Tells if a virtual server is running
           It returns proper exit code, so you can use it as a test
 status  : Tells some information about a vserver
 chkconfig : It turns a server on or off in a vserver

--nodev  : Do not configure the IP aliases of the vserver
           Useful to enter a vserver without enabling its network
           and avoiding conflicts with another copy of this vserver
           running elsewhere
--silent : No informative messages about vserver context and IP numbers
           Useful when you want to redirect the output

mother :~# vserver wonko enter
ipv4root is now 147.162.35.81
New security context is 10
root@wonko:/# df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/hdv1             30984584   2803488  28181096  10% /
```

Vserver: come funziona / 1

- Nuove syscall:
  - **`new_s_context (int ctx)`**
  - Ctx: ID contesto, namespace univoco per i processi di quel contesto e i suoi figli.
  - **`set_ipv4root(unsigned long ip)`**
  - Indirizzo IP immutabile per un dato contesto.
- Fa uso delle **capabilities** per limitare i poteri all'interno di un contesto: `/usr/include/linux/capability.h`

Vserver: come funziona / 2

- 3 nuovi comandi base:
  - `/usr/sbin/chcontext`
  - `/usr/sbin/chbind`
  - `/usr/sbin/reducecap`
- Comandi modificati:
  - `vps, vpstree, vrpm, vkill, vdu, vserver, vhalt, vtop, vserver-stat, newvserver, reducecap ...`
- *Unification*:
  - Per non replicare n volte TUTTI i file

Il file *vserver*.conf

- **IPROOT** (IPROOTDEV, IPROOTMASK, IPROOTBCAST):
  - IPROOT="eth0:192.168.1.2 eth1:192.168.3.1/ 255.255.255.192"
- **ONBOOT**=yes (parte al boot del server ospite)
- **S_CAPS**=CAP_NET_RAW (per il ping & co.)
- **S_DOMAINNAME** = NIS domainname
- **S_HOSTNAME** = hostname
- **S_NICE** = -20 to 19 (livello di priorita' dei processi)
- **S_FLAGS** =lock, sched, nproc, private, fakeinit
- **ULIMIT**="-H -u 1000" (parametri di ulimit)

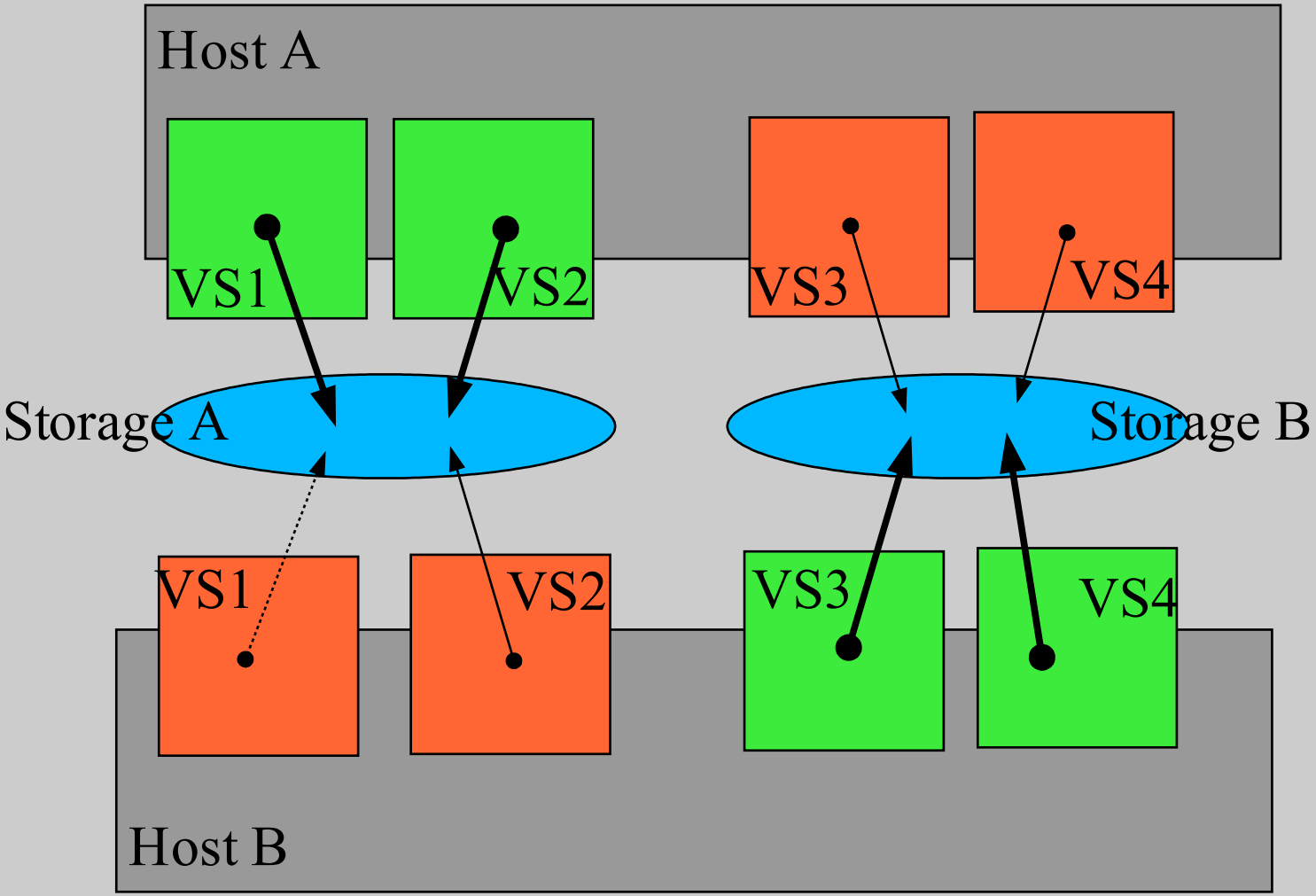
Esempio 1

```
IPROOT=147.162.35.1
IPROOTMASK=255.255.255.0
IPROOTBCAST=147.162.35.255
IPROOTDEV=eth0
ONBOOT=no
S_HOSTNAME=marvin.stat.unipd.it
S_DOMAINNAME=
S_NICE=
# You can set various flags for the new security context
# lock: Prevent the vserver from setting new security context
# nproc: Limit the number of processes in the vserver according to ulimit
#       (instead of a per user limit, this becomes a per vserver limit)
S_FLAGS="lock nproc"
ULIMIT="-H -u 1000"
# You can set various capabilities. By default, the vserver are run
# with a limited set, so you can let root run in a vserver and not
# worry about it. He can't take over the machine. In some cases
# you can to give a little more capabilities (such as CAP_NET_RAW)
S_CAPS="CAP_NET_RAW"
```

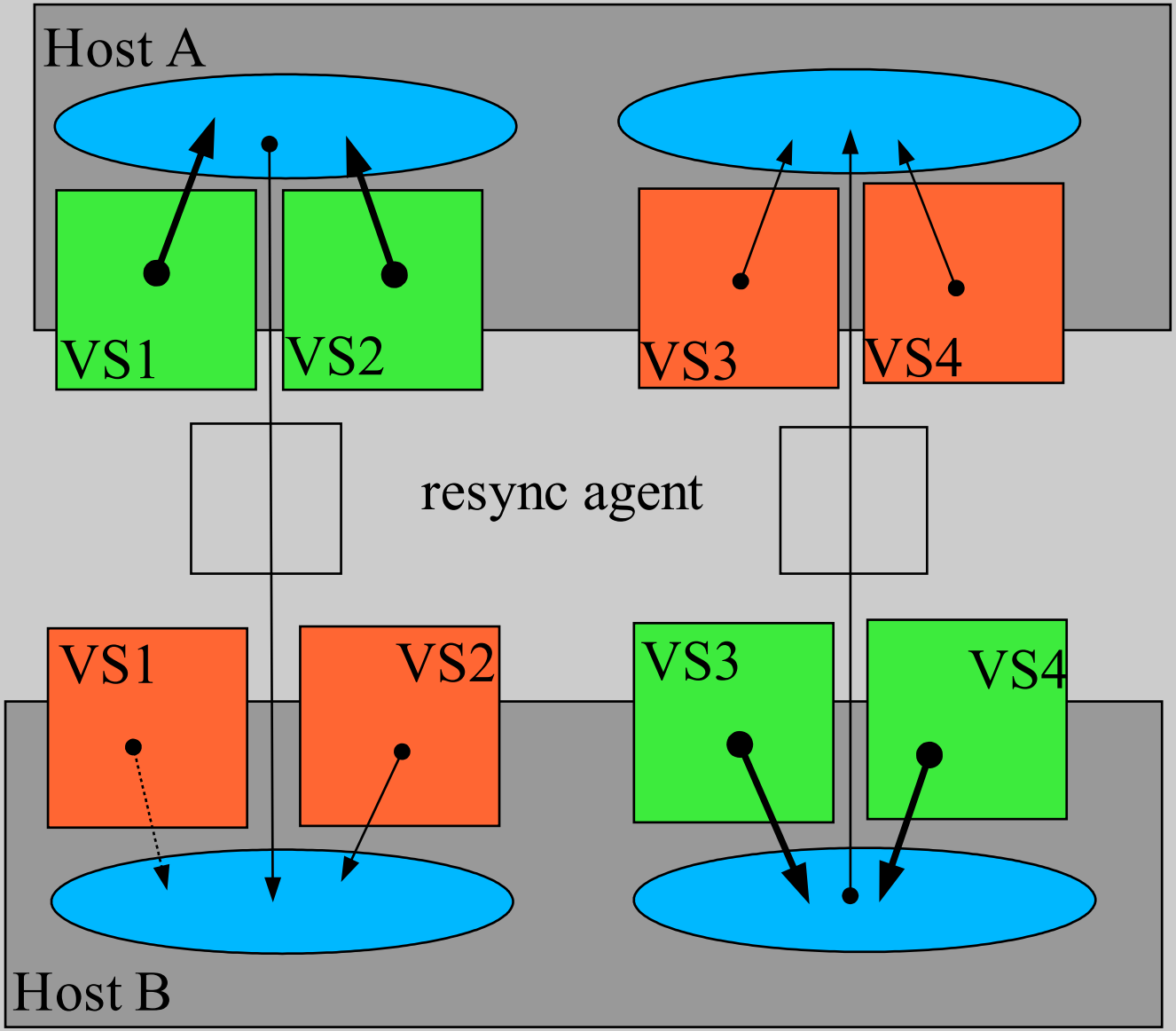
Esempio 2

```
#!/bin/sh
if [ "" = "" ] ; then
        PROFILE=prod
fi
case $PROFILE in
prod)
        IPROOTDEV="eth0"
        IPROOT="147.162.35.206"
        IPROOTMASK="255.255.255.0"
        IPROOTBCAST="147.162.35.255"
        S_HOSTNAME="db"
        S_CAPS="CAP_NET_RAW"
;;
backup)
        #IPROOT=1.2.3.4
        #IPROOTMASK=
        #IPROOTBCAST=
        #IPROOTDEV=eth0
        S_HOSTNAME=
;;
esac
ONBOOT=yes
S_DOMAINNAME=
S_NICE=
S_FLAGS="lock nproc"
ULIMIT="-H -u 1000"
S_CAPS="CAP_NET_RAW"
```

HA per i ricchi (shared storage)



HA per i poveri (resync)



Vserver Webografia

- http:/ / www.linux-vserver.org/
- http:/ / www.paul.sladen.org/ vserver/ howto.html
- ftp:/ / ftp.solucorp.qc.ca/ pub/ vserver
- http:/ / www.solucorp.qc.ca/ miscprj/ s_context.hc
- http:/ / list.linux-vserver.org/ mailman/ listinfo/ vserver
- http:/ / www.13thfloor.at/ vserver
- http:/ / homes.stat.unipd.it/ mmzz/ Papers/ NewVserver/ index.html