

# **T-OSSLab: seminario sulla Virtualizzazione**

# Appunti del laboratorio pomeridiano sugli strumenti di virtualizzazione per Linux e Xen.

## Indice degli argomenti:

<b>1. Indice degli argomenti:</b> .....	<b>2</b>
<b>2. Premessa</b> .....	<b>3</b>
<b>3. Approfondimenti sui risvolti pratici della virtualizzazione</b> .....	<b>3</b>
2.1 Funzionalità .....	3
2.2 Criticità .....	4
2.3 Compatibilità .....	5
<b>4. Laboratorio - Utilizzo dello strumento xen-tools per la creazione di macchine virtuali</b> .....	<b>6</b>
Caso D'uso .....	6
3.1 Situazione AS IS .....	6
3.2 Proposta di intervento.....	6
3.3 Attività esecutive: .....	6
3.3.1 Personalizzazione del file di configurazione di xen-tools.....	7
3.3.2 Creazione della macchina virtuale di test.....	8
3.3.3 Analisi dei log.....	9
3.3.4 Attivazione della macchina virtuale.....	9
3.4 Strumenti di analisi: .....	10
3.5 Conclusioni .....	10
3.5.1 Risultati raggiunti .....	10

## Premessa

Il giorno 19/02/2009, presso il dipartimento di Informatica dell'Università di Pisa, si è tenuto l'incontro pomeridiano sull'argomento della virtualizzazione con Xen. In tale sede, sono stati forniti argomenti di riflessione sui risvolti pratici dalla virtualizzazione descritta nel corso della mattina ed è stata offerta ai partecipanti la possibilità di valutare personalmente la facilità d'uso e la potenzialità dello strumento **xen-tools** mediante prove pratiche al computer.

## Approfondimenti sui risvolti pratici della virtualizzazione

Nella prima parte del pomeriggio è stata offerta ai partecipanti la possibilità di chiedere approfondimenti sui concetti emersi durante la mattina. Di seguito sono riassunte le peculiarità emerse suddivise per tematiche:

- Funzionalità
- Criticità
- Compatibilità

### 2.1 Funzionalità

Domanda: VMware offre la possibilità di spezzare le immagini in tanti file di dimensioni inferiori, con evidenti vantaggi in termini di gestione. Lo stesso vantaggio si può ritrovare con Xen?

Risposta: Sì, sono disponibili strategie diverse.

- nel caso in cui vi sia la possibilità di accedere alla macchina, si possono utilizzare strumenti come 'rsync'.
- è possibile ricorrere all'uso dello strumento 'dd'.
- è possibile ricorrere a vari comandi UNIX.
- esistono servizi di housing che offrono servizi di backup on demand della macchina virtuale.
- in alternativa, è possibile utilizzare il comando UNIX 'tar' che consente di copiare i soli dati senza che sia necessario copiare l'intera struttura del filesystem, con evidenti guadagni in termini di spazio occupato.

Domanda: Quali sono i vantaggi dell'utilizzo di LVM nella creazione di una macchina virtuale? Lo strumento LVM si integra facilmente con la suite xen-tools.

Risposta: I vantaggi legati all'impiego dello strumento LVM, in alternativa ai file sono notevoli, tra i principali:

- Prestazioni: con LVM elimino l'overhead introdotto dall'utilizzo dei file.
- Sicurezza: con LVM è di norma più difficile cancellare una macchina virtuale accidentalmente.
- Scalabilità: LVM è un metodo di allocazione dello spazio del disco fisso in volumi logici che possono essere facilmente ridimensionati al contrario delle partizioni di tipo tradizionale.

Inoltre LVM si integra con xen-tools. L'utilizzo di xen-tools combinato con LVM permette di disporre di molti vantaggi in modo user-friendly.

Domanda: E' possibile impostare la priorità delle macchine virtuali?

Risposta: Xen mette a disposizione degli strumenti per impostare caratteristiche a runtime: 'vcpu-set' 'mem-set'. Ad ogni modo è necessario tener presente che non esistono dei comandi diretti per impostare la priorità: l'hypervisor raccoglie le richieste delle macchine virtuali e gestisce le code mediante algoritmi complessi.

## 2.2 Criticità

Domanda: Quali criticità comporta la virtualizzazione?

Risposta: La virtualizzazione introduce alcune criticità importanti:

- Manutenzione della Macchina fisica: è evidente che la macchina fisica rappresenta un bottleneck per tutte le macchine virtuali che ospita. Pertanto la sua manutenzione deve essere accurata.
- Sicurezza. La paravirtualizzazione è potenzialmente più pericolosa della Full Virtualization. Nella paravirtualizzazione, il kernel della macchina fisica viene modificato in modo da sostituire le operazioni privilegiate con una chiamata all'hypervisor. Questo leggero strato software fa da proxy tra la macchina virtuale e quella fisica. Sebbene l'accesso ai dischi sia emulato, e quindi non sia permesso ad un cracker che abbia preso il controllo di una macchina virtuale cancellare i dischi della macchina fisica ospitante, alcune istruzioni arrivano direttamente al processore, e possono essere utilizzate da un eventuale malintenzionato per rendere instabile il kernel e addirittura di conseguenza predisporre un 'hole' per un futuro accesso alla macchina fisica. Inoltre, attraverso metodi di fingerprinting è possibile ottenere informazioni relative alla macchina fisica ospitante. In conclusione, quando si utilizza lo strumento della paravirtualizzazione è necessario tenere presente che LA SICUREZZA DEL SISTEMA DEVE ESSERE GARANTITA GLOBALMENTE.
- Generazione di Key SSH/SSL. Il livello di sicurezza associato alle chiavi generate su macchine virtuali è molto basso. Il grado di entropia che si ottiene su una macchina virtuale è sempre inferiore a quello garantito da una macchina fisica dove è presente rumore hardware. In generale, le macchine virtuali risultano meno adatte ad essere utilizzate per esperimenti scientifici che necessitino l'impiego di numeri random. Ne consegue, che nei casi in cui risulti peculiare disporre di un elevato livello di entropia, sarà necessario prestare attenzione a quei servizi di hosting basati su infrastrutture virtuali.

Domanda: Riuscendo ad accedere ad una Macchina, è possibile capire se si tratta di una macchina virtuale?

Risposta: Sì, ci sono molti metodi per riuscire a capire su una macchina è o meno virtuale. Di seguito sono riassunti i principali:

- **MAC-ADDRESS.** I primi tre byte del MAC ADDRESS identificano il produttore della scheda di rete. Nel caso di macchine virtuali basate su Xen, I primi tre byte sono univocamente rappresentati con **'00:16:3e'**. Detto questo, è sempre bene ricordare che un eventuale cracker potrebbe rubare l'indirizzo fisico di un'altra macchina, pertanto è necessario prestare attenzione.
- **Processi attivi.** In Linux lo strumento 'ps' e relative opzioni, permette di elencare i processi in esecuzione su una macchina. E' possibile rendersi conto che una macchina è virtuale semplicemente andando ad analizzare l'output di questo comando. Facciamo un esempio: il comando 'ps aux' dato all'interno di una macchina virtuale mostra due processi 'xenwatch', 'xenbus'. La presenza di tali processi fa capire facilmente che la macchina che stiamo analizzando è virtuale. E' doveroso segnalare, comunque che esistono opportune tecniche che permettono di mascherare la presenza di tali processi.
- **Servizi mancanti:** Le macchine virtuali Xen solitamente non supportano direttamente le librerie NPTL (TLS).
- **hardware:** noto l'hardware di una macchina fisica, ed utilizzando tool appropriati di interrogazione, può essere semplice rendersi conto se un sistema è virtualizzato . Facciamo un esempio: utilizzando il tool 'hwclock' per interrogare l'hardware clock su una macchina virtuale, probabilmente otterremo una risposta come di seguito: " Cannot access the Hardware Clock via any known method." In altre parole una macchina virtuale non risponde oppure risponde qualcosa di emulato.
- **Entropia:** come detto al punto precedente, le macchine virtuali non garantiscono un buon livello di entropia. Esistono dei tool per il rilevamento dell'entropia che possono essere utilizzati per valutare se il sistema è virtuale. Per contro è comunque necessario tener presente che esistono dei software che aiutano a generare entropia.

Domanda: Quante interfacce di rete 'Dom0' posso avere?

Risposta: Il numero limite di interfacce di rete non costituisce un problema, in quanto è normalmente difficile raggiungere il limite teorico. Il problema più importante è costituito dalla latenza introdotta dalle code software. In pratica un numero di interfacce virtuali più alto comporta un degrado prestazionale. Un altro aspetto fondamentale è costituito dall'utilizzo della rete. Le interfacce sono configurate in modalità promiscua, le patch applicate al Kernel della macchina fisica, rendono possibile alla stessa di comunicare con diversi MAC address a seconda della macchina virtuale con cui viene stabilita la connessione. E' fondamentale tenere presente che disattivando l'interfaccia fisica non si ha la sicurezza massima relativamente alla disattivazione delle interfacce virtuali, che è bene disattivare di conseguenza.

## 2.3 Compatibilità

Domanda: La distribuzione Slackware può essere riconosciuta dallo strumento xen-tools?

Risposta: xen-tools privilegia quelle distribuzioni con repository accessibili e manutenibili con tool di installazione come Aptitude.

Domanda: Le macchine virtuali possono essere viste come un metodo per continuare ad utilizzare distribuzioni obsolete e/o non più supportate dalla community?

Risposta: in generale, la virtualizzazione può essere vista come possibile soluzione. Ad ogni modo la paravirtualizzazione non è in questo caso facilmente implementabile, in quanto oltre al problema di patchare kernel obsoleti si possono presentare anche problemi di compatibilità e supporto di periferiche hardware.

## **Laboratorio - Utilizzo dello strumento xen-tools per la creazione di macchine virtuali**

Durante la sessione di laboratorio è stata offerta ai partecipanti la possibilità di apprezzare la semplicità della suite xen-tools per la creazione di una macchina virtuale basata su xen.

È stato mostrato un primo caso d'uso a cui è seguita una descrizione dettagliata di tutti gli step seguiti. Quindi è stata offerta ai partecipanti la possibilità di creare per conto proprio una macchina virtuale customizzata secondo gli specifici desideri.

Di seguito i dettagli del caso d'uso visto insieme.

### **Caso D'uso**

#### **3.1 Situazione AS IS**

È stata predisposta una macchina fisica con

- Sistema Operativo: Ubuntu Hardy 8.04 Server LTS
- pacchetto: ubuntu-xen-server
- RAM: 2Gbyte

#### **3.2 Proposta di intervento**

L'obiettivo del presente caso d'uso consiste nella creazione di una macchina virtuale basata su Sistema Operativo Ubuntu Hardy 8.04 Server LTS, utilizzando la suite xen-tools.

#### **3.3 Attività**

Le attività previste sono:

- Personalizzazione del file di configurazione di xen-tools
- Creazione della macchina virtuale di test
- Analisi dei log
- Attivazione della macchina virtuale creata

### 3.3.1 Personalizzazione del file di configurazione di xen-tools

Viene modificato il file di configurazione `'/etc/xen-tools/xen-tools.conf'` in modo da stabilire i parametri di default da passare alla procedura **xen-create-image**.

Al momento della creazione della macchina virtuale, tali parametri potranno comunque essere modificati specificandoli come opzioni del comando **'xen-create-image'**.

Di seguito viene mostrato il file di configurazione di esempio.

-----  
# lvm = gv # # If you don't wish to use loopback images then you may specify an LVM volume group here instead  
**dir = /home/xen** # il parametro 'dir' permette di creare una macchina virtuale mediante file

**install-method = debootstrap**

# Disk and Sizing options. defaults

**size = 1Gb** # Disk image size.

**memory = 128 Mb** # Memory size

**swap = 128 Mb** # Swap size

**fs = ext3** # il parametro fs permette di esplicitare il filesystem che si desidera utilizzare.

**dist = hardy**

**image = sparse** # Specify sparse vs. full disk images.

# Currently supported and tested distributions include:

#

# via Debootstrap:

#

# Debian:

# sid, sarge, etch, lenny.

#

# Ubuntu:

# edgy, feisty, dapper.

#

# via Rinse:

# centos-4, centos-5.

# fedora-core-4, fedora-core-5, fedora-core-6, fedora-core-7

# Networking setup values.

**dhcp = 1** # il parametro dhcp impostato al valore '1' abilita l'indirizzamento dinamico tramite richieste al dhcp.

**gateway = x.x.x.x**

**netmask = x.x.x.x**

**broadcast = x.x.x.x**

# Misc options

**cache = 1** # il parametro cache impostato ad '1', abilita il caching, ottimizzando notevolmente i tempi di download di sistemi operativi precedentemente scaricati mediante il metodo 'debootstrap'

**passwd = '1'** # il parametro 'passwd' abilita la richiesta della password al termine della procedura di creazione della macchina virtuale

```
# Default kernel and ramdisk to use for the virtual servers
```

```
#kernel = /boot/vmlinuz-`uname -r`
```

```
#initrd = /boot/initrd.img-`uname -r`
```

# il parametro kernel è importantissimo, il valore ad esso assegnato deve corrispondere con l'identificativo del Kernel presente sulla macchina fisica. Questo in accordo con il fatto che nella paravirtualizzazione il kernel non è installato. Ne consegue che macchina fisica e macchine virtuali devono essere compatibili!

```
kernel = /boot/vmlinuz-2.6.24-23-xen
```

```
initrd = /boot/initrd.img-2.6.24-23-xen
```

```
# Lo strumento debootstrap consente di scaricare il sistema operativo preferito selezionando il mirror appropriato
```

```
# mirror_sid=http://ftp.us.debian.org/debian
```

```
# mirror_sarge=http://ftp.us.debian.org/debian
```

```
# mirror_etch=http://ftp.us.debian.org/debian
```

```
# mirror_dapper=http://archive.ubuntu.com/ubuntu
```

```
# mirror_edgy=http://archive.ubuntu.com/ubuntu
```

```
# mirror_feisty=http://archive.ubuntu.com/ubuntu
```

```
# mirror_gutsy=http://archive.ubuntu.com/ubuntu
```

```
mirror_hardy=http://archive.ubuntu.com/ubuntu
```

```
# Filesystem options for the different filesystems we support.
```

```
#
```

```
#ext3_options = noatime,nodiratime,errors=remount-ro
```

```
#ext2_options = noatime,nodiratime,errors=remount-ro
```

```
#xfs_options = defaults
```

```
#reiser_options = defaults
```

---

### 3.3.2 Creazione della macchina virtuale di test

Il comando di seguito riportato consente la creazione della macchina virtuale di test, nonché la creazione del file di configurazione `'/etc/xen/test2.cfg'` in modo del tutto trasparente per l'utente.

```
xm-create-image --hostname=test2 --ip=x.x.x.x #al posto di x.x.x.x è stato impostato un indirizzo IP della rete locale.
```

Completato questo passaggio l'utente avrà comunque la possibilità di customizzare la macchina virtuale secondo le specifiche esigenze.

Esempio di file di configurazione xen per la macchina virtuale:

```
# Configuration file for the Xen instance test2, created
```

```
# Kernel + memory size
```

```
#
```

```
kernel = '/boot/vmlinuz-2.6.24-23-xen'
```

```
ramdisk = '/boot/initrd.img-2.6.24-23-xen'
```

```
memory = '128'
```

```
#
# Disk device(s).
#
root    = '/dev/xvda2 ro'
disk    = [
    'tap:aio:/home/xen/domains/test2/disk-img,xvda2,w',
    'tap:aio:/home/xen/domains/test2/disk-swap,xvda1,w'.
]

#
# Hostname
#
name     = 'test2'

#
# Networking
#
vif      = [ 'ip=x.x.x.x,mac=00:16:3e:00:00:01' ]

#
# Behaviour
#
on_poweroff = 'destroy'
on_reboot   = 'restart'
on_crash    = 'restart'

extra = '2 console=xvc0'
```

---

### **3.3.3 Analisi dei log**

Completata la creazione della macchina virtuale è possibile analizzare i dettagli delle operazioni che la suite xen-tools ha eseguito per conto dell'utente consultando il file:

```
'/var/log/xen-tools/test2.log'
```

### **3.3.4 Attivazione della macchina virtuale**

Per far partire la macchina virtuale che abbiamo creato è sufficiente un semplice comando:

```
xm create -c /etc/xen/test2.cfg
```

Per consentire l'avvio automatico della macchina virtuale anche a seguito del re-boot della macchina fisica ospitante è sempre possibile creare un link simbolico come di seguito indicato:

In `-s /etc/xen/test2.cfg /etc/xen/auto/test2.cfg`

### 3.4 Strumenti di analisi:

Gli strumenti di analisi messi a disposizione da xen per l'analisi delle macchine virtuali sono molteplici. Di seguito viene descritto uno tra i peculiari: **'xm list'**.

'xm list' costituisce il modo più rapido e diretto per capire se l'hypervisor è attivo. L'output di questo comando si presenta come indicato:

```
-----  
Name                ID      Mem    VCPUs  State    Time(s)  
Domain-0            0      1893    2      r----- 4943.9  
test2                5      128     1      -b----- 53.1  
-----
```

Name: riporta il nome che si è deciso di assegnare alla macchina virtuale. Nel caso specifico abbiamo: 'Domain-0' che identifica la macchina fisica, 'test2' che identifica la macchina virtuale che abbiamo creato ed avviato.

ID: è l'identificativo associato alla macchina. E' fondamentale prestare attenzione al fatto che il valore 'ID' si incrementa ad ogni re-boot.

Mem: questo parametro identifica quanta memoria è assegnata a ciascuna macchina. Importante osservare che la RAM assegnata alla macchina virtuale viene sottratta a quella fisica. La quantità di RAM disponibile per una macchina fisica costituisce un vincolo alla quantità di macchine virtuali che possono essere create.

VCPU: rappresenta il numero di CPU VIRTUALI assegnate a ciascuna risorsa.

Time: questo parametro è fondamentale, in quanto riporta il Tempo di CPU, ovvero il carico di lavoro della CPU. Durante la sessione di Laboratorio i partecipanti hanno potuto constatare che il Tempo di CPU della macchina fisica aumenta all'aumentare delle macchine virtuali che si vanno a creare, anche se tali macchine non offrono alcun servizio aggiuntivo.

## 3.5 Conclusioni

### 3.5.1 Risultati raggiunti

La prova di laboratorio ha evidenziato la potenzialità e la facilità d'uso dello strumento xen-tools. I partecipanti hanno avuto modo di testare personalmente la rapidità con cui la suite consente di creare una macchina virtuale da zero.