Laboratori didattici informatici multidisciplinari e multifunzionali: un modello architetturale software

Rocco Alibertini, Massimiliano Ciancio[1]
Università degli Studi "G. D'Annunzio" Chieti Pescara
Via dei Vestini 31, 66100 Chieti (CH)
Università Telematica "L. Da Vinci"
Piazza San Rocco, 66010 Torrevecchia Teatina (CH)
r.alibertini@unidav.it
[1]*ITCGT "F. Galiani"*
via Ricci 22, 66100 Chieti (CH)
massimiliano.ciancio@istruzione.it

Questa esperienza progettuale nasce dall'esigenza di ottimizzare l'uso dei laboratori in un istituto tecnico commerciale. Nella pratica comune capita spesso che i laboratori scolastici siano inutilizzabili a causa della presenza di virus, di modifiche volontarie o involontarie della configurazione iniziale dei PC o per la non capillare installazione dei software su tutti i computer. Con questo lavoro si intende proporre una soluzione alle criticità esposte e ad altre innumerevoli criticità che si presentano continuamente durante un anno scolastico. Come? Presentando un modello la cui implementazione trasformi i laboratori scolastici in multidisciplinari e multifunzionali, ma soprattutto che limiti al minimo la manutenzione software. A quali costi? I costi sono minimi e si riferiscono solo all'implementazione del modello, grazie al ricorso al sorprendente mondo del software libero. Il modello è esportabile in una qualunque altra scuola di ogni ordine e grado nonché nelle università.

1. Introduzione

Se si facesse una statistica sull'uso dei laboratori nelle scuole italiane, probabilmente emergerebbe che molti di essi vengono scarsamente utilizzati, soprattutto nelle scuole primarie e secondarie di primo grado: la motivazione è che, per la maggior parte dell'anno scolastico, i laboratori sono resi indisponibili da problemi software. Indagare le cause alla base di questi problemi non è obiettivo del presente lavoro, ma certamente qualche indicazione al riguardo, basata sull'esperienza sul campo, può essere data: le cause più comuni sarebbero da ricercare nella mancanza di fondi per la manutenzione, nella mancanza di personale in grado di effettuarla e nelle debolezze intrinseche al sistema operativo Windows, solitamente utilizzato nei laboratori scolastici.

A. Andronico, A. Labella, F. Patini (Eds.): DIDAMATICA 2010 – ISBN 978-88-901620-7-7

È possibile ridurre al minimo la manutenzione software? Questo lavoro propone un modello che, se implementato, permette di ridurre a due gli interventi di macro-manutenzione nell'arco di un anno scolastico e al minimo (tendente allo zero) gli interventi di micro-manutenzione, laddove per macro-manutenzione si intende la contemporanea attività su tutte le macchine del laboratorio o dei laboratori e per micro-manutenzione ci si riferisce ad interventi spot su singole macchine.

2. Obiettivi

L'esperienza sul campo ed una tendenziosa generalizzazione rendono credibile l'assunto che, nella maggior parte delle scuole italiane, la dotazione tecnologica sia insufficiente.

Nelle scuole più tecnologiche, di contro, si osserva che i laboratori didattici sono altamente specializzati, ad esempio laboratorio di lingue accessibile solo ai linguisti, laboratorio CAD accessibile solo agli ingegneri, laboratorio di informatica accessibile solo agli informatici e così via. Analizzando le differenze strutturali di tali laboratori ci si accorge che essi differiscono solo per qualche periferica hardware e per il software installato sui computer. Accade, quindi, che tale ricchezza tecnologica diventi comunque insufficiente: le varie configurazioni specialistiche, che intimoriscono gli insegnanti meno esperti, nonché la convinzione dei responsabili di laboratorio che l'eccessiva frequentazione degli stessi sia correlata ad un aumento della quantità dei problemi e ad una diminuzione della loro affidabilità, impediscono, di fatto, il più ampio accesso possibile ai laboratori.

Rimuovere questo ostacolo è il primo obiettivo che questo lavoro si propone: indicare un layout software indistinto ed utilizzabile da qualunque disciplina. Da qui l'idea di rendere intercambiabili e multifunzionali i laboratori e di facilitarne l'accesso.

Per centrare questo obiettivo bisogna creare un'offerta multipiattaforma realizzata con strumenti per la virtualizzazione. In questo modo ogni insegnante, avendo a disposizione più sistemi, può scegliere semplicemente quello più aderente alle sue esigenze e, volendo, può decidere di passare da un sistema all'altro senza tempi morti. Tale obiettivo potrebbe essere raggiunto anche con il partizionamento dei dischi e l'installazione dei sistemi in partizioni diverse, con il problema, però, dei lunghi tempi di attesa nel passaggio da un sistema all'altro. Ricorrendo alla virtualizzazione, invece, è possibile risolvere questi inconvenienti e creare, in teoria, su una singola macchina infinite macchine virtuali [Moriggia, 2009].

L'altro obiettivo da raggiungere è l'affidabilità delle singole macchine. L'uso continuo, soprattutto da parte di utenti sempre diversi, rende le macchine indifese rispetto a configurazioni personali introdotte dagli utenti, a modifiche della configurazione iniziale, ad installazioni di software non attinenti alla didattica e, infine, ad attacchi di malware.

Il modello proposto rende di fatto i laboratori altamente affidabili e a prova di manomissione. L'esperienza biennale di laboratori basati su tale modello ci

permette di affermare che l'utilizzabilità e l'affidabilità sono caratteristiche ormai assunte in maniera definitiva.

3. Il modello

Il modello prevede l'utilizzo di:

- un **server d'istituto** sul quale configurare il servizio FTP (File Transfer Protocol) per offrire un repository centralizzato di file. Ad ogni studente e ad ogni docente viene assegnata una *Cartella personale* a cui si può accedere in modo sicuro attraverso una coppia formata da *username* e *password.* In questo modo è possibile lavorare in uno qualunque dei laboratori dell'istituto avendo sempre a disposizione i propri file ed evitando che gli studenti possano, intenzionalmente o per errore, cancellare o modificare i file di altri studenti. Nel nostro caso è stato utilizzato un server Linux Ubuntu 8.04 con il demone *vsftpd* (FTP server) attivo [Canonical, 2008];
- un **server d'aula** sul quale configurare i servizi FTP, DHCP (Dynamic Host Configuration Protocol), SSH (Secure SHell) e sul quale far risiedere alcuni script pensati per la didattica quali *internet_on/off* e *lock_on/off*. L'aula è configurata come una sottorete, in modo da alleggerire il carico di lavoro della rete d'istituto. Il servizio DHCP assegna gli indirizzi IP ai client d'aula secondo uno schema prestabilito. L'installazione di un server FTP d'aula si rende necessario per l'utilizzo del software G4U (Ghosting For Unix) descritto in seguito. Utilizzando il programma ClusterSSH, che sfrutta a sua volta il protocollo SSH, è possibile collegarsi contemporaneamente ai singoli client d'aula e svolgere interventi di micro-manutenzione in parallelo su tutte le postazioni presenti. Lo script *lock_on/off* è stato realizzato in modo che dalla postazione docente sia possibile bloccare l'utilizzo delle macchine client durante le spiegazioni. Lo script *internet_on/off* è stato realizzato in modo che dalla postazione docente sia possibile attivare la navigazione Internet sulle macchine client solo quando necessaria. Nel nostro caso è stato utilizzato il sistema operativo Linux Ubuntu 9.04 con i demoni *vsftpd*, *dhcp3-server* (DHCP server), *openssh* (Secure SHell protocol) attivi;
- **client d'aula** sui quali installare e configurare un sistema host Linux, un software per la virtualizzazione (VirtualBox), un sistema virtuale Windows e alcuni script per garantire l'affidabilità delle postazioni. La scelta di configurare i client d'aula con Linux deriva dal fatto che quest'ultimo è un sistema operativo libero, più affidabile e meno vulnerabile di Windows. VirtualBox, anch'esso software libero, è stato scelto per la sua semplicità d'uso. L'implementazione della macchina virtuale Windows, invece, è richiesta dall'enorme diffusione che caratterizza questo sistema operativo, dalle competenze già in possesso dai docenti e dal fatto che alcuni software proprietari non sono disponibili per Linux. Sui client d'aula sono stati utilizzati alcuni script pensati per garantire l'affidabilità delle postazioni. In particolare lo script *reset_utente,* che viene lanciato in automatico al termine di ogni sessione di lavoro, provvede a ripristinare le impostazioni della directory */home/utente* e a riportare la macchina virtuale Windows alla situazione salvata nell'ultimo *snapshot* (istantanea) dall'amministratore, abbandonando così tutte le modifiche

effettuate dagli utilizzatori dei client. Complementare allo script *reset_utente* sono stati approntati altri script per effettuare modifiche alla configurazione e salvarne il nuovo stato: si tratta degli script *blocca_utente*, *sblocca_utente* e *salva_utente.* Nel nostro caso, per i client d'aula, sono stati utilizzati il sistema operativo Linux Ubuntu 9.04 con i demoni *vsftpd* e *openssh* attivi, VirtualBox [Sun Microsystems, 2010], Windows XP e vari software didattici sia per Linux che per Windows.

In basso si riporta uno schema grafico riassuntivo del modello (vedi Fig. 1).

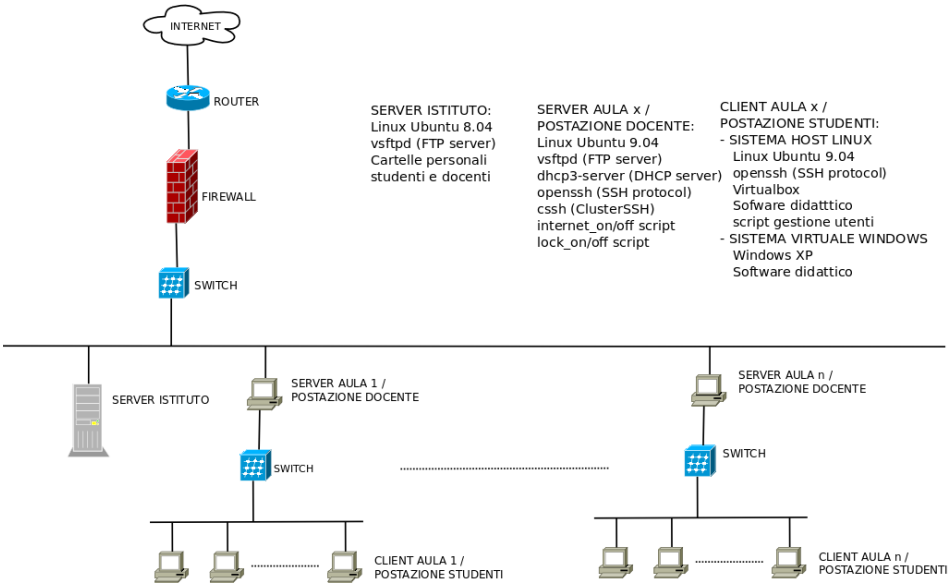

Fig. 1 – Schema del modello

4. L'implementazione del modello

L'implementazione del modello non può prescindere da un'analisi di partenza che abbia per oggetto sia la dotazione hardware e software dell'istituto, sia le esigenze software degli insegnanti utilizzatori dei laboratori. Dal risultato di questa analisi discendono le varie fasi dell'implementazione che vanno dalla creazione dei master, all'installazione dei vari server.

4.1 L'analisi della dotazione hardware e software

Il censimento dell'hardware deve restituire informazioni che permettano di raggruppare i computer in gruppi, ossia di individuare gruppi di computer che hanno lo stesso hardware e per i quali sia possibile un'unica configurazione software, da clonare successivamente su ogni macchina.

È altresì importante conoscere le licenze software possedute per individuare eventuali vincoli, che si scontrano sia con le esigenze economiche della scuola, sia con le esigenze didattiche degli insegnanti.

4.2 L'analisi del fabbisogno software degli insegnanti

All'analisi oggettiva dell'hardware e del software, deve seguire un'analisi delle esigenze didattiche degli insegnanti. Questi ultimi, rispondendo ad un questionario relativo alle loro necessità software e prendendo visione delle disponibilità, indicano le loro preferenze.

4.3 La creazione dei master

L'implementazione del modello inizia con l'individuazione di una macchina da configurare per ogni gruppo di PC identici. Questa macchina verrà poi replicata utilizzando G4U su tutte le postazioni con identico hardware.

Si procede con l'installazione e la configurazione del sistema operativo Linux Ubuntu. La configurazione deve prevedere la creazione di un utente standard, che verrà utilizzato da studenti e docenti, sul quale intervenire ad ogni fine sessione con lo script *reset_utente*, in modo da ripristinare la configurazione base: così facendo ogni azione degli utenti viene abbandonata e la micro-manutenzione sul sistema Linux si azzera; il sistema risulta inattaccabile e aperto a qualunque tipo di esercitazione didattica. Si prosegue poi con l'installazione di tutti i programmi didattici, quindi, con l'installazione di VirtualBox.

Successivamente, occorre creare la macchina virtuale Windows, installandovi il sistema operativo e i relativi software didattici. Una volta impostata correttamente per la connessione ad Internet, nella macchina virtuale va creata una cartella condivisa con il sistema host Linux sottostante: questa soluzione è molto utile nel caso sia necessario lavorare contemporaneamente con entrambi i sistemi e si abbia l'esigenza di condividere rapidamente dei file. Le cartelle condivise rappresentano il modo più semplice e veloce in VirtualBox per condividere i file tra i sistemi; occorre ricordare, però, che essa rappresenta una soluzione di memorizzazione temporanea, poiché, al termine della sessione, ogni modifica viene eliminata.

L'ultimo passo da eseguire per terminare l'installazione del master prevede la creazione di uno snapshot del sistema virtuale Windows da ripristinare al termine di ogni sessione tramite lo script *reset_utente*: in questo modo, ripristinando lo snapshot originario, ogni azione degli utenti viene abbandonata e la micro-manutenzione sul sistema virtuale Windows si azzera, rendendolo sicuro al 100% contro virus o manomissioni; è possibile, così, eseguire qualunque tipo di esercitazione sulla macchina virtuale, anche a livello di sistema, senza pericolo di comprometterla.

4.4 L'utilizzo di G4U

Il modello prevede l'utilizzo del software G4U, che consiste in una distribuzione NetBSD minimale che può essere installata e avviata da CD. G4U permette di salvare/ripristinare una partizione o un intero disco su/da un server

FTP, acquisendo all'avvio un indirizzo IP da un server DHCP. Nel nostro caso G4U permette di salvare l'immagine del computer master sul server d'aula e poi lanciare in parallelo il suo ripristino su tutte le restanti postazioni.

Questa procedura consente di avviare in 15-20 minuti il ripristino di un intero laboratorio (macro-manutenzione) che si conclude, senza ulteriore intervento tecnico, in circa due ore.

4.5 Gli script

Nell'implementazione del modello descritto sono stati realizzati ed utilizzati diversi script sia lato server d'aula che lato client d'aula.

4.5.1 Script per la gestione dell'utente

Servono a creare la corretta configurazione dell'utente standard sia sulla macchina host Linux che sulla macchina virtuale Windows ed a ripristinarla correttamente al termine di ogni sessione:

- **reset_utente**: dopo avere verificato la presenza del file *reset_utente.flag,* utilizzato per attivare/disattivare la funzionalità di reset, provvede a ripristinare la home dell'utente sulla macchina host Linux in base alla configurazione salvata ed a ripristinare la sessione salvata sulla macchina virtuale Windows; lo script *reset_utente* viene inserito all'interno del file */etc/gdm/PostSession/Default* che viene invocato al termine di ogni sessione utente:

```
#!/bin/bash
# controlla che sia presente il file reset_utente.flag:
# se non esiste esce e non effettua il reset
if [ ! -e /root/bin/reset_utente.flag ] ; then
exit;
fi
# ripristina la home utente
rm -rf /home/utente
tar xzf /opt/home_utenti/utente.tar.gz -C /
chown -R utente.utente /home/utente
# ripristina sessione VirtualBox scartando le modifiche e tornando
allo snapshot salvato su - utente -c "VBoxManage snapshot winxp
discardcurrent -state"
```

- **attiva_reset_utente**: crea il file *reset_utente.flag* utilizzato come flag; la sua assenza indica che non si deve effettuare il reset al termine di ogni sessione:

```
#!/bin/bash
touch /root/bin/reset_utente.flag
```

- **disattiva_reset_utente**: elimina il file *reset_utente.flag* utilizzato come flag; la sua assenza indica che non si deve effettuare il reset al termine di ogni sessione:

```
#!/bin/bash
rm -f /root/bin/reset_utente.flag
```

- **salva_utente**: salva il contenuto della home utente sulla macchina host Linux in modo da poterlo poi ripristinare all'invocazione dello script *reset_utente*:

```
#!/bin/bash
rm -rf /opt/home_utenti/utente.tar.gz
tar czf /opt/home_utenti/utente.tar.gz /home/utente
```

- **inizia_modifiche_windows**: consente di effettuare modifiche sulla macchina virtuale Windows che verranno poi rese permanenti tramite lo script *termina_modifiche_windows*:

```
#!/bin/bash
/root/bin/disattiva_reset_utente
# elimina lo snapshot
su utente -c "VBoxManage snapshot winxp discardcurrent -all"
```

- **termina_modifiche_windows**: conclude le modifiche alla macchina virtuale Windows, salvando lo *snapshot* che verrà poi ripristinato al termine di ogni sessione utente tramite lo script *reset_utente*:

```
#!/bin/bash
/root/bin/attiva_reset_utente
# scatta uno snapshot
su utente -c "VBoxManage snapshot winxp take ok"
echo "ATTENDERE PREGO......!"
/root/bin/salva_utente
echo "FINITO :)"
```

4.5.2 Script per la didattica d'aula

I seguenti script sono stati pensati per migliorare la didattica d'aula. Consentono di gestire la navigazione Internet, l'accesso degli studenti ai computer e l'utilizzo della macchina virtuale Windows. Sono stati integrati nella barra degli strumenti sul desktop per garantirne l'utilizzo a tutti i docenti:

- **internet_off**: agendo sul *forwarding* del server d'aula, disabilita l'inoltro dei pacchetti verso Internet, consentendo solo l'inoltro verso l'indirizzo IP del server d'Istituto:

```
#!/bin/bash
iptables -D FORWARD -i eth1 -j ACCEPT
iptables -A FORWARD -i eth1 -d 192.168.1.2 -j ACCEPT
```

- **internet_on**: agendo sul *forwarding* del server d'aula, riabilita l'inoltro dei pacchetti verso qualsiasi destinazione ed elimina la regola, superflua, che consente l'accesso verso il server d'Istituto:

```
#!/bin/bash
iptables -D FORWARD -i eth1 -d 192.168.1.2 -j ACCEPT
iptables -A FORWARD -i eth1 -j ACCEPT
```

- **all_halt**: lanciata come utente *root*, consente di spegnere le postazioni studente dalla postazione docente; utilizza il comando *ssh* per eseguire il comando *halt* su tutte le postazioni:

```
#!/bin/bash
RETE='10.0.0'
HOST=`seq 2 26`
for X in $HOST ; do
ssh $RETE.$X "halt" &
done
```

- **all_start_winxp**: consente di avviare dalla postazione docente tutte le macchine virtuali sui client d'aula; utilizza il comando *ssh* per eseguire il comando d'avvio sulle postazioni studente:

```
#!/bin/bash
RETE='10.0.0'
HOST=`seq 2 26`
for X in $HOST ; do
ssh  $RETE.$X  "DISPLAY=':0.0'  /usr/local/bin/StartWinXP  remote"  &>
/dev/null &
```

```
done
```

all_stop_winxp: consente di spegnere dalla postazione docente tutte le macchine virtuali sui client d'aula; utilizza il comando *ssh* per eseguire il comando di chiusura sulle postazioni studente:

```
#!/bin/bash
RETE='10.0.0'
HOST=`seq 2 26`
for X in $HOST ; do
ssh $RETE.$X "VBoxManage controlvm winxp poweroff" &> /dev/null &
done
```

- **lock_on**: consente di bloccare le postazioni studente dalla postazione docente; chiede se si vogliono bloccare tutte le postazioni o solo un singolo computer; per effettuare il blocco lancia sulle postazioni studente, attraverso *ssh,* una versione modificata e ricompilata dello scrensaver *xhost* in cui è stata disabilitata la terminazione; lo studente, dunque, non potrà sbloccare la postazione perché non avrà più accesso alla tastiera e al mouse e soltanto il docente, lanciando lo script *lock_off* potrà permettere agli studenti di utilizzare i computer:

```
#!/bin/bash
RETE='10.0.0'
HOST=`seq 2 26`
XLOCK='/usr/local/bin/xlock'
SINGLE=`zenity --entry --title="Blocco postazioni" --text
"Postazione da bloccare (0 per tutte le postazioni)" --entry-
text="0"`
if [ $SINGLE -gt 0 ] ; then
HOST=$SINGLE
fi
for X in $HOST ; do
ssh $RETE.$X "DISPLAY=':0.0' $XLOCK -mode marquee -message
'postazione bloccata ...'" &> /dev/null &
done
```

- **lock_off**: termina, attraverso *ssh*, l'esecuzione dello screensaver modificato *xlock*, riassegnando agli studenti il controllo delle postazioni; chiede se si vuole sbloccare una singola postazione o l'intera aula:

```
#!/bin/bash
RETE='10.0.0'
HOST=`seq 2 26`
SINGLE=`zenity --entry --title="Sblocco postazioni" --text
"Postazione da sbloccare (0 per tutte le postazioni)" --entry-
text="0"`
if [ $SINGLE -gt 0 ] ; then
HOST=$SINGLE
fi
for X in $HOST ; do
ssh $RETE.$X "killall -9 xlock" &> /dev/null &
done
```

5. La manutenzione

La manutenzione software che potrebbe rendersi necessaria in laboratori basati sul modello proposto può consistere in:

- manutenzione periodica svolta su tutte le macchine con l’obiettivo di aggiornare il software in uso (macro-manutenzione);

- manutenzione necessaria per ripristinare singoli client, da effettuare saltuariamente solo in presenza di problemi software (micro-manutenzione).

5.1 La macro-manutenzione

Una prima macro-manutenzione va fatta all'inizio dell'anno scolastico per risolvere gli eventuali problemi sorti alla fine dell'anno scolastico precedente e, soprattutto, per aggiornare i vari pacchetti software installati: tale operazione consiste nell'aggiornamento del client master, nella creazione della sua immagine attraverso G4U e nella sua successiva clonazione sugli altri client.

Normalmente non sono necessari altri interventi di macro-manutenzione; in situazioni particolari, tuttavia, dati i tempi brevi che necessitano per il ripristino dell'intero laboratorio, tali interventi potrebbero essere tranquillamente effettuati anche durante l'anno senza prevedere blocchi della didattica laboratoriale. Si consiglia un secondo intervento durante la sospensione delle attività didattiche prevista nel periodo natalizio.

5.2 La micro-manutenzione

La micro-manutenzione va effettuata ogni volta che si verifica un problema software di qualunque tipo sulla singola macchina. Questi interventi, come già detto, si riducono drasticamente con l'applicazione del modello. Durante il secondo anno di sperimentazione si sono registrati, in media, soltanto quattro interventi di micro-manutenzione per laboratorio. Considerando il numero dei laboratori esistenti nell'istituto di applicazione ed il loro intenso utilizzo, si può affermare che le attività laboratoriali non hanno mai avuto sosta, con un incredibile vantaggio per la didattica.

Tra gli interventi di micro-manutenzione si possono annoverare:

- il ripristino da immagine di un singolo client d'aula;
- l'intervento in parallelo attraverso ClusterSSH su tutti i client per modifiche alle configurazioni o installazioni di nuovi software sull'host Linux;
- eventuali modifiche alla macchina virtuale Windows, replicate sui client d'aula attraverso la copia via SCP (Secure CoPy) dei file di VirtualBox.

6. Conclusioni

L'applicazione del modello proposto ha permesso di trasformare i laboratori dell'istituto in ambienti multidisciplinari e multifunzionali in maniera trasparente per insegnanti e alunni. Essa ha permesso inoltre di utilizzare i laboratori in maniera indistinta ed intensiva, rimuovendo tutte le resistenze dei docenti, legate alla convinzione che esista una correlazione tra quantità degli accessi e numerosità dei problemi che si manifestano. La sperimentazione biennale effettuata negli anni scolastici 2007/2008 – 2008/2009 presso l'ITCGT "F. Galiani" di Chieti, ci permette di affermare che il layout software proposto rende di fatto i laboratori altamente affidabili e a prova di manomissione. Eventuali problemi, inoltre, possono essere risolti immediatamente considerando il fatto che un intero laboratorio può essere ripristinato in un paio d'ore.

Bibliografia

[Canonical, 2008] Canonical, Ubuntu Server Guide, Canonical Ltd., London, 2008.

[Moriggia, 2009] Moriggia F., Computer infiniti. PC Professionale, 225, 12, 2009, 32-49.

[Sun Microsystems, 2010] Sun Microsystems, Sun VirtualBox User Manual, Sun Microsystems, Inc., San Francisco, 2010.

Sitografia

[http://sourceforge.net/projects/clusterssh/], sito ufficiale del progetto del software libero ClusterSSH.

[http://www.feyrer.de/g4u/], sito ufficiale del software libero Ghosting For Linux.

[http://www.fsf.org/], sito ufficiale della Free Software Foundation.

[http://www.ubuntu.com/], sito ufficiale del sistema operativo libero Linux Ubuntu.

[http://www.virtualbox.org], sito ufficiale del software libero VirtualBox.