

Note specifiche per la versione distribuita tramite Internet

Negli ultimi giorni di preparazione alla laurea – consegna del "nullaosta" del relatore, stampa in copisteria, consegna del libretto, timbratura delle copie e via dicendo -, croce e delizia di ogni laureando, e' prevista anche la dichiarazione per l'autorizzazione alla consultazione della propria tesi da parte di altri studenti. Il laureando e' libero di decidere se permettere tale pratica o meno.

Non ho mai avuto dubbi in proposito: avevo la chiara intenzione di permettere a *chiunque* (dal piu' antipatico compagno di corso al piu' spregevole dei professori) di consultare il mio lavoro, qualora lo ritenesse degno di nota, interessante e degno del piu' spregiudicato dei "copia e incolla". E se permettevo al mio compaesano fiorentino di leggerla, perche' non al barese? o al romano? e all'*italian-speaker* di New York?

Da questa riflessione all'upload su Internet sono passati pochi attimi.

Ho deciso quindi di pubblicare su Internet la mia tesi di laurea, in Gestione Informatica dei dati aziendali, discussa nel luglio 2005. Il tema è "*La migrazione dei sistemi da piattaforma Windows a Linux*".

Questa è la versione integrale dell'intero elaborato, frutto di studio, ricerca e passione, e il risultato di mesi di lavoro.

Questo documento è distribuito con licenza Creative Commons, piu' precisamente la "Attribution-NonCommercial-ShareAlike 2.0". Per poterla rispettare in tutti i suoi aspetti, ti consiglio di leggerne la spiegazione su <http://creativecommons.org/licenses/by-nc-sa/2.0/> .

Non esitare a contattarmi per qualsiasi dubbio in merito al mio lavoro

ed alla licenza CC da me scelta.

La tesi, o meglio *questa* tesi, non e' in nessun modo da considerarsi un documento "statico": il desiderio di rimetterci le mani e correggere, modificare, migliorare/peggiore e' frequente cosi' come il changelog. Tutto dipende dal tempo a disposizione per farlo. Attualmente sto lavorando alla versione in html dell'intero testo.

Anche per questo motivo sono particolarmente ben accette critiche e consigli sul tema e sul lavoro in generale.

Il mio indirizzo email è, attualmente :), antani@gmail.com

Buona lettura!

orangeek



UNIVERSITA' DEGLI STUDI DI FIRENZE

FACOLTA' DI ECONOMIA

CORSO DI LAUREA IN *ECONOMIA E COMMERCIO*
TESI DI LAUREA IN *GESTIONE INFORMATICA DEI DATI*
AZIENDALI

**"LA MIGRAZIONE DEI SISTEMI DA PIATTAFORMA
WINDOWS A LINUX"**

Relatore: Chiar.mo

Il Candidato

Prof.ssa *Maria Grazia Ceppatelli*
Docente in *Gestione informatica*
dei dati aziendali

orangeek

ANNO ACCADEMICO 2004/2005

Alla mia famiglia

Ringraziamenti

Ringrazio i miei genitori e mio fratello per avermi sopportato in tutto questo tempo. Ringrazio la Professoressa Maria Grazia Ceppatelli per la disponibilità e l'interesse dimostratomi. Non posso non essere grato ai miei amici (*tutti*) per avermi sopportato in questi anni, durante la stesura di questo lavoro e soprattutto durante le mie, improvvisate, mai richieste e logorroiche, arringhe in difesa e sostegno, di Linux e del software libero: credo che mi chiederò sempre se il loro annuire fosse per effettivo accordo con le mie opinioni o semplicemente per terminare l'agonia. Sono anche particolarmente riconoscente a Francesco, che “in tenera età” mi consigliò Linux e che solo dopo qualche anno ho ascoltato. Ringrazio Tex per avermi fatto compagnia nelle pause dallo studio. Ringrazio poi la comunità del software libero, italiana ed internazionale: se non fosse stato per loro, molto probabilmente non avrei mai conosciuto, ormai qualche anno fa, la "rivoluzione più importante nel mondo del software".

La migrazione dei sistemi da piattaforma

Windows a Linux

Indice

Ringraziamenti.....	4
Introduzione.....	9
1. Origine e diffusione dei due sistemi operativi.....	23
1.1 Le origini di Microsoft Windows.....	23
1.2 Le origini di GNU/Linux.....	33
1.2.1 Il Free Software.....	36
1.2.2 Free Software e software Open Source: le ambiguità.....	41
1.3 GNU/Linux, un mondo a se stante.....	43
1.4 Windows: un'ascesa inarrestabile.....	46
2. Perché migrare.....	50
2.1 Il problema della Sicurezza Informatica: panoramica	50
2.1.1 Chiarezza sul tema della sicurezza informatica.....	62
2.1.2 Windows e Linux “in the wild”	66
2.1.3 Rigorosa gestione della multiutenza.....	74
2.1.4 Distribuzione di patch e bugfixing.....	78
2.1.5 Sicurezza nel browser.....	84
2.1.6 Modularità del kernel.....	86
2.2 Flessibilità	89
2.3 Facilità nella manutenzione dei sistemi.....	92
2.4 Riduzione dei costi legati ai client Linux.....	96
2.4.1 Costi di licenza per il sistema operativo.....	97
2.4.2 Costi di licenza per il software aggiuntivo.....	101
2.4.3 Costi per l'acquisizione di hardware.....	105
2.4.4 Costi per la manutenzione dei client	107
2.5 Vendor tie-in.....	110
3. Analisi e pianificazione preliminare.....	115
3.1 Analisi preliminare.....	116
3.2 Distinzione in thin e fat client	118
3.2.1 Thin Client.....	119
3.2.2 Fat Client.....	120
3.3 Ruolo del client ed applicazioni utilizzate.....	121
3.4 Attuali problemi ed esigenze delle postazioni	124
3.5 Quando non migrare.....	125
3.6 Utilizzo di applicazioni web based.....	130
3.6.1 Costi di licenza: scenario attuale e rischi per il futuro.....	130
3.6.2 Fat client e sicurezza.....	134
3.6.3 I software groupware.....	136
4. Strumenti e tecnologie per la migrazione.....	140
4.1 Applicazioni.....	140
4.1.1 Assenza di funzioni: il caso Microsoft Office.....	141
4.1.2 Assenza dell'alternativa software per Linux	146
4.1.3 Bridging Application.....	149

4.1.4	Utilizzo dell'applicazione tramite Terminal Server.....	155
4.1.5	Le alternative software.....	159
4.2	Interfaccia utente.....	165
4.2.1	Interfaccia grafica e aspetto del desktop.....	168
4.2.2	Differenze terminologiche e percorsi dei file.....	171
4.3	Server e infrastruttura di rete.....	175
4.3.1	Convidisione in rete di file e directory.....	179
4.3.2	Condivisione di stampanti.....	181
4.3.3	Server DHCP e DNS.....	185
4.3.4	Proxy Server.....	187
4.4	Processo di migrazione pilota.....	189
4.4.1	Applicazioni	192
4.4.2	Sistema operativo	194
4.4.3	Infrastruttura.....	195
4.4.4	Terminal Server.....	197
4.4.5	Raccolta del feedback e assistenza tecnica.....	198
4.5	Migrazione completa.....	199
4.5.1	Fattore Umano.....	202
4.5.2	Formazione.....	206
5.	Punti deboli di Linux e soluzioni.....	210
5.1	Compatibilità hardware.....	212
5.2	Difficoltà di installazione, configurazione e manutenzione.....	218
5.3	Difficoltà nell'utilizzo.....	225
5.3.1	Installazione di applicazioni tramite codice sorgente.....	227
5.3.2	Accesso ai filesystem.....	233
5.4	Scarsità di applicazioni.....	237
5.5	Interoperabilità con altri sistemi operativi.....	240
6.	Un'applicazione pratica: Schwaebisch Hall.....	243
6.1	Applicazioni pratiche nel mondo aziendale.....	244
6.2	L'amministrazione comunale di Schwaebisch Hall.....	247
6.2.1	Nascita del progetto “Linux im Rathaus”.....	248
6.2.2	Motivazioni.....	249
6.2.3	Condizioni da rispettare per la migrazione	255
6.2.4	Pianificazione della migrazione.....	257
6.2.5	Implementazione tecnica.....	258
6.3	La migrazione delle applicazioni specifiche.....	262
6.4	Realizzazione della migrazione.....	266
6.5	Conclusioni.....	269
	Conclusioni.....	272
	Bibliografia.....	278

Introduzione

Il mondo dell'Information Technology è in continua evoluzione: costantemente nuovi operatori compaiono nel settore nel volgere di pochi mesi e, altrettanto velocemente, rischiano di scomparire per sempre a causa dell'incessante innovazione apportata dai concorrenti. La rapidità con la quale vengono adottate nuove tecnologie e nuovi strumenti informatici non permette a nessuno di rimanere indietro o di non informarsi in merito.

Lo scenario dovrebbe essere caratterizzato da un'assenza pressoché totale di certezze: non esiste una tecnologia che, nel volgere di brevissimo tempo, non rischi l'obsolescenza ed il rapido abbandono.

I cambiamenti avvenuti nel corso degli anni Novanta del secolo scorso sono stati importanti e rapidissimi. Il personal computer è, ormai, uno strumento utilizzato ed utilizzabile da chiunque, nelle quotidiane attività, nel mondo del lavoro e non solo. Lo strumento informatico, in precedenza ad esclusivo appannaggio di utenti esperti, è diventato oggi di uso comune e familiare alla maggior parte della popolazione. Il merito di questa evoluzione è da attribuire senza dubbio anche agli operatori e agli sviluppatori che, consci delle potenzialità dello strumento, hanno reso il sistema operativo ed i suoi applicativi sempre più semplici da usare, più potenti ed allo stesso tempo più intuitivi.

Attualmente il mercato è dominato da un'unica importante azienda:

Microsoft. Sviluppatrice del sistema operativo Windows, diffuso in varie versioni, fornisce i propri prodotti a più del novanta per cento degli utenti di personal computer. Forse il merito maggiore nella enorme diffusione del personal computer è imputabile a questa azienda statunitense. Grazie al proprio sistema operativo, ha reso lo strumento informatico utilizzabile e sfruttabile da chiunque. “*Desktop*” è diventato un termine di uso comune, Microsoft Office la suite di produttività personale più diffusa al mondo.

Microsoft nel volgere di pochissimi anni ha affrontato e battuto importanti avversari sul mercato; Apple e IBM hanno dovuto "semplicemente" accettare la vittoria dell'azienda di Bill Gates che ha conquistato l'intero settore. In una realtà sempre più “connessa” e ricca di network, dominata dalla grande rete Internet, Microsoft ha cominciato a sviluppare una versione del sistema operativo Windows dedicata al mercato “*server*”. I suoi prodotti, inizialmente dedicati all'utente comune ed all'utilizzo personale di un computer, vengono ora installati anche sui server, macchine atte a prestare servizi di rete per una moltitudine di postazioni. Microsoft, pur inserendosi in un mercato più maturo e con attori stabilmente inseriti nel suo interno, riesce in ogni modo ad imporsi ed i prodotti dedicati ai propri sistemi acquistano rapidamente un'importante diffusione.

Dopo anni di dominio del mercato di un unico operatore, quale può essere lo spazio per un valido concorrente? Quale azienda potrebbe mai fronteggiare un simile avversario?

La proposta avanzata nel nostro lavoro non consiste in un prodotto sviluppato da un'azienda, ma da una comunità di programmatori: *GNU/Linux*.

Questo sistema operativo deve il suo nome ai suoi due sviluppatori principali: la *GNU Foundation* fondata da Richard Stallman ed uno studente finlandese all'epoca diciannovenne, Linus Torvalds. Le innovazioni più importanti apportate da Linux non sono state esclusivamente tecniche: è, infatti, il primo sistema operativo completamente *libero*.

Chiunque può studiarne il codice, la sua “ricetta segreta”, modificarlo e farne l'uso che preferisce. Migliaia di sviluppatori di differenti nazionalità hanno così collaborato a questo progetto, assicurando al sistema operativo libero un'importante e rapida evoluzione. Linux ha mostrato al mondo dell'Information Technology che il software a sorgente libero può raggiungere ottimi livelli qualitativi, che lo sviluppo basato sull' *open source* è possibile e che il mercato del software non necessita esclusivamente di importanti investimenti, ma soprattutto di idee, capacità e volontà.

A metà degli anni Novanta GNU/Linux era ancora un sistema operativo complesso, utilizzabile esclusivamente da utenti esperti e caratterizzato da problemi e difficoltà che rendevano il suo uso inavvicinabile ai più. Negli ultimi anni, però, la comunità del software libero ha conosciuto importanti

novità: il codice di Linux è diventato più stabile, gli sviluppatori si sono dedicati alla semplificazione dell'interfaccia e ad una maggior intuitività del software, sono nati nuovi progetti al fine di rendere il sistema operativo quanto più completo ed immediato possibile, mantenendo allo stesso tempo le caratteristiche di stabilità, sicurezza e flessibilità alla base del suo processo di sviluppo. L'obiettivo era inevitabilmente raggiungere il livello di alcune caratteristiche di Microsoft Windows: l'occasione più importante era conquistare l'utente comune e rendere il sistema operativo libero utilizzabile da chiunque.

Non è passato molto tempo ed i frutti di questo impegno sono rapidamente comparsi: importanti enti pubblici hanno manifestato vivo interesse nei confronti di Linux ed hanno cominciato ad implementare le soluzioni basate sull'open source all'interno delle proprie infrastrutture. Grandi aziende del settore, come IBM e Novell, hanno investito importanti risorse per la sua diffusione, convinte così di poter affrontare ad armi pari il dominatore del mercato.

Linux costituisce ormai una reale alternativa ai prodotti sviluppati da Microsoft, potenzialmente capace di apportare importanti vantaggi e benefici a chi deciderà di implementarlo.

Cambiare sistema operativo ed applicativi costituisce, però, una scelta importante e da valutare attentamente: la diffusione di Windows e dei suoi applicativi ha avuto rilevanti effetti sul mercato e sugli utenti e non

considerare adeguatamente simili aspetti è un rischio che non deve essere corso in nessun caso.

Il nostro lavoro affronterà il tema della migrazione dei sistemi informatici, all'interno di una realtà aziendale, da piattaforma Microsoft Windows al sistema operativo Linux. A causa della complessità del processo e delle importanti conseguenze che ne possono derivare, è stata posta la massima attenzione a tutte le fasi della migrazione.

Nel corso del nostro lavoro affrontiamo in primo luogo una panoramica sullo sviluppo e sulla diffusione dei due sistemi operativi, prestando particolare attenzione al software libero per chiarire le differenze concettuali che distinguono il *free software* e l'*open source* dal software proprietario.

Successivamente saranno analizzate le motivazioni alla base della decisione di migrare i propri sistemi da piattaforma Windows a Linux, illustrandone i benefici sotto diversi punti di vista. Particolare attenzione sarà dedicata al tema della *sicurezza informatica* che, negli ultimi anni, ha raggiunto un'importanza vitale; dopo l'esposizione delle ripercussioni tecniche ed economiche più gravi, illustreremo come, a nostro avviso, Linux costituisca, in confronto a Windows, un sistema più sicuro ed affidabile. Affronteremo questa tematica sotto diversi aspetti, concentrandoci su quelli di maggior rilievo per una realtà aziendale interessata all'implementazione di Linux.

La sicurezza del sistema operativo libero, però, non costituisce l'unica motivazione per la migrazione dei sistemi; importanti fattori come la *flessibilità* del sistema e del software e la *facilità di manutenzione* dei sistemi possono aumentare notevolmente i vantaggi conseguibili tramite l'implementazione di Linux nella propria realtà aziendale, garantendo la possibilità di concentrare le proprie risorse su altri aspetti.

Particolare attenzione, poi, sarà dedicata alle *riduzione dei costi* legata alla scelta di Linux e del software libero come principale risorsa dei propri sistemi informatici. L'argomento verrà affrontato in base ai diversi capitoli di spesa, comprendenti l'acquisto delle licenze del sistema operativo, del software aggiuntivo e di terze parti, per l'acquisizione dell'hardware e per la manutenzione dei sistemi. In un momento di crescente innovazione nelle tecnologie e di necessità di importanti competenze, questi fattori giocano un ruolo determinante nella gestione di una realtà aziendale. Affronteremo a riguardo anche la questione delle politiche di licenza d'uso praticate nel settore del software proprietario ed i rischi ad esse connesse in vista delle importanti innovazioni che stanno avvenendo nel settore.

Scegliere il software open source significa poi *libertà*. In un mercato dominato da un unico attore, le aziende che facciano uso intenso di software proprietario possono solo subire passivamente le decisioni e le strategie impiegate dal proprio fornitore: in questo settore, poi, non è concesso rimanere indietro e non aggiornarsi. Uno scenario di questo tipo costituisce

un fattore estremamente rischioso per le aziende-clienti, costrette a sopportare gli effetti di un grave “*vendor tie-in*”, che pone loro in una situazione di pesante, e rischiosa, dipendenza dal fornitore del software. L'implementazione del software open source e di Linux annullano i rischi per la dipendenza da un singolo fornitore: l'intero sistema operativo e tutte le applicazioni sono liberamente disponibili e la comunità del software libero è, in quanto a supporto, una risorsa inesauribile. Le uniche decisioni da rispettare ed esigenze da soddisfare sono le *proprie*.

Alla luce di queste motivazioni, riteniamo che la migrazione dei sistemi a piattaforma Linux sia estremamente vantaggiosa per un'azienda; con il nostro lavoro, però, andiamo oltre il semplice progetto di implementazione del software open source sulle macchine *server*, come spesso suggerito nel settore. Queste, per definizione, costituiscono solo una minima frazione di un network aziendale: l'obiettivo più importante è quello di effettuare la migrazione sulle postazioni *client*, costituenti la maggior parte dei sistemi. L'installazione di Linux su queste macchine agisce come importante *moltiplicatore dei benefici* apportabili dal software open source in azienda.

Conoscere quali siano i vantaggi derivanti dall'implementazione di Linux non è sufficiente: la migrazione dei sistemi informatici è un processo complesso. L'importanza delle fasi iniziali di *analisi preliminare e pianificazione* è fondamentale: un'errata previsione delle conseguenze della migrazione può essere causa di insuccesso. Una configurazione superficiale

dei sistemi può rendere il sistema non usabile e può creare resistenze da parte degli utenti. Sottovalutare il *fattore umano* della migrazione, con tutte le inevitabili ripercussioni sugli utenti finali, può comportare il fallimento dell'intero processo o il sostenimento di importanti spese per correggere decisioni prese in precedenza. Nel corso del nostro lavoro dedicheremo particolare attenzione a queste due fasi iniziali della migrazione, analizzando approfonditamente tutti gli aspetti ad esse legati.

L'implementazione di Linux, pur essendo potenzialmente molto vantaggiosa, può non essere la soluzione adatta in particolari situazioni: un recente aggiornamento dei sistemi basandosi su software proprietario o particolari configurazioni delle postazioni costituiscono, ad esempio, scenari per i quali la decisione di migrare a Linux deve essere accuratamente ponderata e valutata. Tratteremo a riguardo i casi più esemplificativi, proponendo anche una possibile configurazione per network aziendali tramite l'utilizzo di applicazioni “*web based*” al fine di esaltare i vantaggi in termini di sicurezza, efficienza e riduzione dei costi di licenza.

Queste fasi preliminari, fondamentali per l'intero processo, sono il preludio al momento dell'*implementazione pratica* di Linux: la nostra attenzione si sposterà, quindi, sugli aspetti tecnici di questa fase.

Il *cambiamento delle applicazioni* utilizzate costituirà l'ostacolo più difficile da superare: il nostro lavoro proporrà e valuterà le alternative più valide

offerte dal software libero. Obiettivo della migrazione non deve essere esclusivamente il raggiungimento dei vantaggi indicati in precedenza, ma deve anche escludere l'interruzione delle attività o la diminuzione della produttività delle postazioni facenti parte il network aziendale. La notevole diversità che queste ultime possono presentare rispetto ad altre configurazioni, poi, rende impossibile proporre un'unica strategia implementativa per la migrazione delle applicazioni. Talvolta, sostituire i software utilizzati non è possibile mantenendo un'alta produttività delle postazioni; talvolta le applicazioni utilizzate in precedenza con Microsoft Windows sono effettivamente necessarie e non è possibile trovare alternative nel software libero. In determinate situazioni, che esemplificheremo nel corso della nostra trattazione, effettuare la migrazione completa potrebbe anche comportare il mantenimento di alcune postazioni con Windows o con software proprietario ad esso dedicato; per queste ragioni illustreremo le tecniche e le strategie più importanti in questo ambito, rendendo possibile affiancare le originali applicazioni per Windows alle nuove postazioni Linux.

La migrazione completa dei sistemi, interessando soprattutto le macchine client utilizzate dagli utenti finali, ha inevitabili ripercussioni su questi ultimi: nelle fasi di analisi, pianificazione e implementazione non deve mai essere sottovalutato l'aspetto costituito dal *“fattore umano”*. Alla luce dell'importanza rivestita da questo punto, illustriamo quali siano gli aspetti

tecniche di implementazione e configurazione dei sistemi più adatti ai diversi contesti: la nostra attenzione sarà rivolta sia all'impostazione generale dei sistemi che alla sua interfaccia utente, strumento fondamentale per garantire un produttivo ed immediato utilizzo del personal computer, illustrando le notevoli possibilità ed alternative offerte dal software open source.

Per quanto il nostro lavoro sia concentrato principalmente sulle postazioni desktop degli utenti finali, l'obiettivo principale della migrazione, rivolgeremo la nostra attenzione anche alle macchine server, sempre più diffuse ed importanti all'interno di qualsiasi network aziendale. Frequentemente sono proprio questi sistemi a rappresentare il primo *target* delle “timide” migrazioni a Linux, al fine di usufruire dei vantaggi apportabili dal software open source, senza però influenzare gli utenti finali delle postazioni. Quanto proposto dal nostro lavoro può avvenire senza difficoltà di sorta anche in contesti di rete gestiti da server Microsoft; a nostro avviso, l'implementazione di Linux anche su queste macchine può garantire, come esposto in precedenza, ulteriori vantaggi in termini di costi ed efficienza. L'aspetto che approfondiremo sarà quello della loro configurazione tramite prodotti open source al fine di permettere la compatibilità completa, anche in un contesto di reti miste composte da client Windows e Linux.

L'intero processo di migrazione, poi, sarà trattato distinguendo due importanti fasi di realizzazione: la “*migrazione pilota*” e la *migrazione*

completa di tutte le postazioni. La prima è, a nostro avviso, un passaggio fondamentale, al pari dell'analisi e pianificazione preliminare esposta precedentemente, per poter correggere eventuali errori e per una maggior consapevolezza della portata della migrazione e delle sue ripercussioni. In proposito, il nostro lavoro affronterà tutti gli aspetti fondamentali della migrazione pilota: dalla configurazione delle postazioni alla formazione degli utenti, dalla raccolta del *feedback* alla gestione dell'infrastruttura di rete. Il passo conclusivo sarà la fase finale dell'intero processo, momento nel quale Linux sarà implementato sulla totalità delle postazioni client. Anche a tal fine, indicheremo come la raccolta del *feedback* e la *formazione degli utenti* raappresentino due momenti fondamentali per il successo dell'intero processo e per la valutazione di eventuali correzioni da apportare.

Ogni sviluppatore di software sa che scrivere il codice di un'applicazione “*perfetta*” è impossibile, come ogni altra attività umana. I sistemi operativi, frutto dell'opera di numerosi sviluppatori, confermano questo principio e Linux non sfugge a questa regola. Ai fini di una più esaustiva e completa trattazione, indicheremo quali sono, a nostro avviso, i *difetti* principali del sistema operativo libero, nell'ottica di quanto proposto dal nostro lavoro, e quali le possibili *soluzioni* per affrontarli.

Analogamente, tratteremo anche alcuni punti sui quali, spesso, non vi è chiarezza: la rapidità nell'evoluzione di Linux impone un costante

aggiornamento su quale sia lo stadio di sviluppo e quali le caratteristiche ed i difetti più importanti. Faremo un punto della situazione attuale, coincidente con un'importante fase di maturità dello sviluppo del sistema operativo libero.

La validità e l'efficienza offerta da Linux e dal mondo open source è anche “confermata” dal crescente interesse manifestato da aziende private ed enti pubblici: i vantaggi, esposti in precedenza, hanno reso il sistema operativo libero interessante anche per soggetti “non addetti ai lavori”. I progetti open source più importanti ricevono importanti finanziamenti da aziende private che hanno investito molto su Linux, come IBM e Novell, e da enti pubblici, come l'Unione Europea e la National Security Agency statunitense. Alla luce di questo, concluderemo il nostro lavoro con la trattazione dell'esperienza maturata dall'amministrazione comunale di una città tedesca, Schwaebisch Hall, che ha effettuato la migrazione completa a Linux.

La nostra attenzione si è rivolta a questo caso, piuttosto che all'applicazione pratica in un'azienda privata, al fine di mostrare la validità e la flessibilità mostrata da Linux e dal software open source in scenari particolarmente complessi. I network aziendali, infatti, presentano esigenze e caratteristiche che possono divergere sensibilmente gli uni dagli altri: l'analisi di un'esperienza di questo tipo avrebbe necessariamente orientato l'esposizione a strategie ed a tecniche di implementazione specifiche per la singola realtà aziendale, limitando quindi l'applicabilità in altri contesti.

L'esperienza dell'amministrazione comunale di Schwaebisch Hall, invece, mostra tutta la complessità che una struttura del genere comporta: un numero alto di postazioni, appartenenti a diversi dipartimenti, specializzate per diverse mansioni e con tipologie di utenti diverse. Le applicazioni utilizzate in questi contesti differiscono notevolmente le une dalle altre, variando dai software utilizzati comunemente, come ad esempio Microsoft Office, a programmi sviluppati ad hoc per il comune tedesco. Alla luce di queste premesse e barriere alla migrazione, efficacemente risolte dal dipartimento IT di Schwaebisch Hall, riteniamo che l'esposizione dell'esperienza della città tedesca possa mostrare efficacemente la flessibilità e le qualità di Linux e del software open source.

1. Origine e diffusione dei due sistemi operativi

1.1 *Le origini di Microsoft Windows*

La prima versione di Microsoft Windows comparve nel 1985, versione 1.0. Questa versione non può essere considerata un sistema operativo completo, perché in realtà si trattava di una estensione di Ms-Dos, del quale ne condivideva le funzionalità, così come problemi e limiti. La stessa dotazione software del sistema operativo non comprendeva nessuna applicazione che potesse interessare l'utenza aziendale e questo portò ad una modestissima diffusione del prodotto. Ulteriori limiti di questa versione erano legati a problemi di natura giuridica; le cause legali vertevano sull'utilizzo, non autorizzato, di Microsoft dell'interfaccia grafica creata e sviluppata da Apple Computer. A causa di ciò, ad esempio, le finestre dei programmi potevano apparire esclusivamente affiancate (“*tiled*”) sullo schermo, non consentendo quindi la sovrapposizione (“*overlapped*”). Allo stesso modo, Windows 1.0 non poteva utilizzare la funzione di "Cestino" (*Trashcan*), in quanto Apple sosteneva di detenerne i diritti. Dette limitazioni furono rimosse dalle successive versioni del sistema operativo di Microsoft, grazie alla sottoscrizione di un contratto di licenza con Apple Computer¹.

¹ In una recente intervista (Occorsio, E., *Affari & Finanza, La Repubblica*, 17 gennaio 2005), Steve Jobs, CEO di Apple, ha dichiarato che se la Apple non avesse stipulato l'accordo di licenza in questione (da lui definito, “il peggior contratto mai sottoscritto dall'azienda”), “molto

Circa due anni dopo uscì Windows 2.0; la nuova versione ebbe molto più successo della precedente, soprattutto grazie alle *killer application*² di Microsoft, Word e Excel, che da quel momento esercitarono il loro inarrestabile effetto con tutte le successive versioni del sistema operativo. In realtà era possibile eseguire detti applicativi anche con Ms-Dos; Windows, infatti, veniva di fatto “lanciato” all'avvio e poi chiuso all'uscita dei programmi stessi.

Nel 1990 Microsoft ottenne un notevole successo con la pubblicazione di Windows 3.0. Con l'introduzione della memoria virtuale³, con il miglioramento del rudimentale multitasking e delle applicazioni native, il sistema operativo riuscì a conseguire notevoli consensi.

A breve, l'azienda di Redmond (cittadina nei dintorni di Seattle, USA) rilasciò una versione potenziata e destinata anche all'intrattenimento multimediale, “Windows 3.0 with Multimedia Extension 1.0”: il kit comprendeva una scheda sonora e un lettore Cd-ROM. Con Windows in

probabilmente i sistemi Windows non sarebbero mai esistiti”. Il responsabile di detto accordo, Sculley, ex dirigente di Pepsi-Cola, è stato sempre estremamente criticato da Jobs. Fu proprio quest'ultimo, suo amico, a volerlo nella sua azienda che aveva già conosciuto un notevole successo (la Apple inventò il primo personal computer). Jobs mantenne la carica di presidente e Sculley quella di CEO; dopo pochi anni, le divergenze tra i due, profondamente diversi e caratterizzati da un backstage culturale agli antipodi, divennero insanabili e Steve Jobs fu costretto ad abbandonare la propria azienda. Rientrò, riassumendo tutte le cariche di vertice, solo nel 1997, conseguendo notevoli successi con prodotti come IMac, Ibook e Ipod.

- 2 Un'applicazione in grado di garantire, da sola, il successo di una piattaforma. L'intera suite Microsoft Office (comprendente Word e Excel) è considerata la killer application per Windows; l'importanza assunta da questo pacchetto software, soprattutto per l'utenza aziendale, rendeva quasi secondaria la scelta del sistema operativo di supporto; considerando inoltre che la suite in questione era inizialmente disponibile solo per Windows, si comprende facilmente l'importanza che ha rivestito questo prodotto nella diffusione del sistema operativo Microsoft.
- 3 Strumento utilizzato dai sistemi operativi al fine di simulare una quantità di memoria centrale maggiore di quella fisicamente installata sulla macchina. Solitamente vengono utilizzate porzioni dello spazio disco, di gran lunga maggiore della RAM, per poter immagazzinare informazioni da usare in tempo reale. In ambiente Linux, ad esempio, viene utilizzato lo “*swap*” per compiere questo tipo di operazioni.

queste due versioni, grazie alle sue caratteristiche tecniche, al costo ridotto ed alle minime esigenze hardware rispetto ai prodotti concorrenti, Windows 3.0 fu caratterizzato da una rapida diffusione. Più di dieci milioni di copie furono vendute in meno di due anni e la vendita del proprio sistema operativo divenne per Microsoft la maggior fonte di reddito.

Gli anni successivi furono caratterizzati da un'importante collaborazione tra IBM e Microsoft per lo sviluppo comune di un sistema operativo (OS/2); i due produttori collaborarono per circa cinque anni, ma alla fine l'accordo si ruppe. IBM continuò ad investire sullo sviluppo di OS/2, le cui caratteristiche erano indubbiamente innovative rispetto a quanto era stato scritto fino a quel momento, mentre Microsoft utilizzò il *know-how* sviluppato in questo progetto al fine di scrivere il proprio Windows NT, il sistema operativo professionale che presto sarebbe venuto alla luce.

Nonostante queste incertezze nello sviluppo della collaborazione con IBM, l'azienda di Redmond non trascurò certo il suo Windows 3.0, che venne sostituito dalla versione 3.1 e, poco dopo, 3.11. La minaccia da scongiurare era, all'epoca, l'imminente uscita del "rivoluzionario" OS/2 di IBM. Le armi con le quali combatteva Microsoft fino a quel momento erano state le specifiche tecniche richieste da Windows (più parche rispetto al prodotto concorrente), da un lato, e, dall'altro, la forte diffusione che il proprio sistema operativo aveva già conosciuto. Con la nuova versione, inoltre, l'azienda di Redmond offriva un netto miglioramento nella gestione dei

network e l'introduzione dei font scalabili *True Type* che divennero rapidamente lo standard per i programmi di grafica editoriale e di elaborazione testi.

Nel frattempo, OS/2 e Windows NT vennero alla luce. I pregi ed i difetti dei due sistemi operativi erano sostanzialmente i medesimi; confrontando i due prodotti, la collaborazione degli anni precedenti tra i due produttori di software era palese. Entrambi offrivano funzionalità che nessun altro sistema operativo era in grado di garantire. I problemi, però, erano i medesimi: le richieste hardware erano molto esose all'epoca e la gestione dei *device driver* da parte dei costruttori di hardware era molto complessa. Le case produttrici di hardware erano così riluttanti nell'investire su tali prodotti e preferivano dedicarsi al mercato, più strettamente *desktop*⁴, di Windows 3.11. Nonostante la diffidenza nei confronti della versione NT, prodotto dotato di caratteristiche valide per l'implementazione come server in una LAN, Microsoft si trovò di fronte ad un aumento consistente delle installazioni del proprio sistema operativo, per quanto nella sua variante più semplice e meno innovativa. Le Windows API⁵ divennero lo standard *de facto* per lo sviluppo di software.

4 Nel lavoro si farà riferimento ad una distinzione in base ad un utilizzo "desktop" (o client, in un contesto di rete) e "server" delle macchine. Con il primo si intende l'utilizzo del personal computer in un ambiente sostanzialmente isolato, utilizzando suite di programmi come Microsoft Office ed altri applicativi. E', indubbiamente, l'utilizzo più frequente che gli utenti fanno di un computer. Il "server", invece, trova la sua ragion d'essere in un ambiente di rete, in un network: in questo caso si distingue tra macchine che prestano un servizio ad altre, i server appunto, e i client che ne usufruiscono. I server con i quali gli utenti normali entrano più spesso in collegamento, sono ad esempio i server web ed i server per la posta elettronica.

5 Application Program Interface. Costituiscono le interfacce di programmazione di software per un sistema operativo. Possono essere considerate le "regole base" che uno sviluppatore deve seguire per scrivere un applicativo. Nel caso più frequente (*Java* di Sun costituisce un'eccezione), sono specifiche per ogni sistema operativo.

L'azienda di Redmond, nel frattempo, si preparava a commercializzare Windows 95, probabilmente la versione di maggior successo del proprio sistema operativo. I punti chiave nello sviluppo sono concentrati in nuova interfaccia grafica, più moderna e meno limitata della precedente, il kernel a 32 bit e il supporto per un efficiente multitasking, rendendo possibile l'esecuzione di più applicazioni contemporaneamente. Per quanto riguarda il kernel, Microsoft fu costretta ad abbandonare il progetto iniziale di una nuova versione a 32 bit; esigenze di retrocompatibilità, performance e di mercato (la scrittura *ex novo* del codice avrebbe concesso troppo tempo ad IBM per rilanciare il suo OS/2) spingono il produttore a scrivere solo parte del codice a 32 bit e lo costringono ad importare vecchie porzioni di codice dalle precedenti versioni. Si ritiene che una delle cause principali dei problemi di stabilità ed efficienza di Windows sia proprio il costante e reiterato (anche nelle versioni più moderne) “riutilizzo” di codice obsoleto all'interno dei nuovi prodotti.

IBM, nel frattempo, aveva sviluppato le ultime versioni di OS/2, arrivando alla versione cosiddetta “*Warp*”. I problemi delle versioni precedenti, in termini di esigenze hardware e velocità, sono sostanzialmente risolti e rendono il prodotto IBM adatto all'utenza desktop e server, guadagnandosi una piccola porzione del mercato dei personal computer.

Nell'agosto del 1995 fa però ingresso nel mercato Windows 95 e si impone definitivamente su OS/2, soprattutto con le versioni successive (OSR 2 con

decisi miglioramenti nella stabilità, con l'inserimento del supporto USB per le periferiche e il filesystem FAT32). Nei due anni successivi Microsoft si occupa sostanzialmente degli aggiornamenti per il proprio sistema operativo e nello sviluppo del nuovo prodotto, Windows 98. Questo è da molti considerato più come un importante aggiornamento al vecchio kernel di Windows 95 che come un nuovo sistema operativo. Al di là delle considerazioni sulla novità del codice usato da Microsoft, le modifiche più importanti sono l'inclusione di nuovi driver hardware (garantendo al prodotto la totale compatibilità con le periferiche del periodo, grazie alla sostanziale assenza di alternative nel campo dei sistemi operativi nonché grazie ad una agguerrita campagna di accordi commerciali tra lo sviluppatore e i costruttori hardware), il pieno supporto del filesystem FAT32 e, successivamente, l'implementazione di "Internet Connection Sharing", sul quale torneremo successivamente. Windows 98 è stato anche il prodotto che ha causato a Microsoft le prime serie cause legali per violazione delle normative *antitrust*. Fu IBM, infatti, a porre per primo la questione: Internet Explorer, il browser web di Microsoft, era stato infatti pesantemente integrato nel codice di Windows 98, rendendo, di fatto, impossibile l'utilizzo del sistema operativo senza il browser in questione ed, allo stesso modo, estremamente difficile la proposta di un navigatore web concorrente.

Nonostante i problemi legali che hanno quasi portato l'azienda di Redmond

sull'orlo di una separazione in diverse divisioni operative, sentenza poi ribaltata dalla Corte Suprema che comunque condannò l'azienda ad un risarcimento di 750 milioni di dollari, Windows 98 continuava a guadagnare significative quote di mercato, diventando il sistema operativo più diffuso. La maggior parte dei personal computer venduti presentava Windows preinstallato, praticamente tutte le periferiche e i componenti hardware erano venduti con i soli driver⁶ per Windows 98. Un'ulteriore funzionalità che permise a questo sistema operativo di imporsi fu l'“Internet Connection Sharing”, con il quale diventava possibile la condivisione di un'unica connessione ad internet tra più personal computer⁷. Questa possibilità, di particolare gradimento più per le aziende e per gli uffici che per l'utenza desktop casalinga, costituì un'ulteriore spinta per l'affermazione del sistema operativo in ambito aziendale.

Gli anni successivi nell'attività di Microsoft sono caratterizzati dallo sviluppo delle nuove versioni di Windows, Windows 2000, ME ed XP. Il primo, da molti considerato il miglior sistema operativo dell'azienda,

6 Software utilizzato dal sistema operativo per controllare (“pilotare”, da qui il termine inglese) le periferiche hardware. Fatta eccezione per alcuni dispositivi molto semplici, o con specifiche di pubblico dominio (i modem analogici ne sono un esempio), è impossibile usare hardware senza i relativi driver, scritti dal costruttore. L'unica possibilità, in caso di assenza di supporto da parte della casa madre, è il *reverse engineering* del dispositivo. Consiste nel provare, empiricamente e con apposite applicazioni *debugger*, a comprendere il funzionamento interno del dispositivo ed a scriverne ex novo i driver. Tecniche di questo tipo, che solo talvolta si concludono con del codice paragonabile a quello del produttore, sono la norma per gli sviluppatori di software libero, impossibilitati, altrimenti, all'utilizzo di determinate soluzioni hardware e software.

7 In realtà, la funzionalità “introdotta” da Microsoft con questa denominazione non è altro che il NAT, Network Address Translation, “traduzione degli indirizzi di rete”. Una tecnica, di solito utilizzata da firewall o router, al fine di modificare l'indirizzo IP di un pacchetto in transito. Nel caso di Windows 98, sarebbe più corretto quindi riferirsi al “supporto” per questa tecnologia che ebbe il sistema operativo durante il suo sviluppo.

riuniva in sé le potenzialità per essere un'ottima soluzione, sia dal lato desktop che dal lato server (il prodotto, infatti, fu distinto nella versione *Professional* e *Server*). La facile interfaccia di Windows 98 (alla quale la maggior parte degli utenti era ormai abituata) insieme a nuove funzionalità più al passo con i tempi e grazie ad un kernel solido, da molti considerato il più stabile di Redmond, rese possibile la diffusione di Windows 2000 in tutti i mercati, dall'utenza aziendale alla sempre più diffusa utenza domestica.

Contemporaneamente all'uscita di questo sistema operativo, Microsoft pubblicò anche Windows ME, *Millenium Edition*; in realtà, si trattava di una versione di Windows 98, leggermente modificata e resa più adatta alle funzionalità multimediali che il mercato richiedeva. Il risultato fu deludente: la maggior parte dell'utenza, preoccupata della diffusa instabilità di questo sistema, preferì optare tra mantenere installato Windows 98, ovvero passare al nuovo 2000.

Microsoft raggiunge un importante traguardo nella strategia di sviluppo nel 2001 quando viene rilasciato Windows XP. Con questo prodotto viene finalmente raggiunta la convergenza delle linee di prodotto dei sistemi operativi che Microsoft da tempo cercava⁸. Windows XP costituisce un prodotto molto discusso sia sul fronte della stabilità che su quello della sicurezza; riguardo quest'ultima, le perplessità sono legate alle numerose

⁸ Questo obiettivo fu perseguito anche con Windows 2000, prodotto che si rivelò valido sia per l'utenza desktop che per quella server. Il codice del sistema operativo, però, divergeva sensibilmente tra le due versioni *Professional* e *Server*; nel caso di Windows XP, invece, vi è una condivisione importante del codice tra le varie versioni.

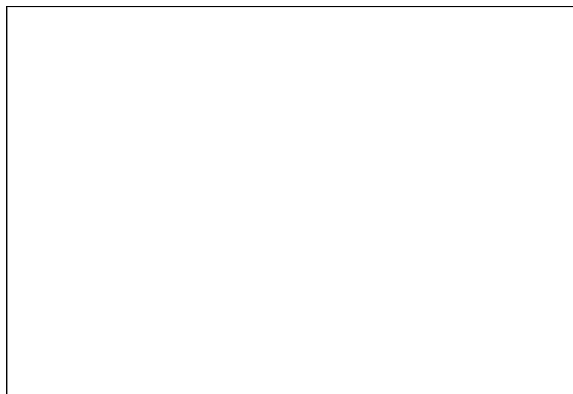
falle di sicurezza presenti nel sistema operativo e nei suoi componenti principali. Queste hanno reso drammaticamente rapida ed “efficiente” la diffusione di “malware”⁹, sotto forma di *worm* e *virus*¹⁰, tramite Windows, Internet Explorer e Outlook Express, il client di posta elettronica di Microsoft, che negli ultimi anni hanno creato seri danni ad importanti strutture informatiche nel mondo. Anche nel caso di Windows XP, così come fu per 2000, Microsoft offre una versione desktop ed una server, quest'ultima sostituita nel 2003 da Windows Server 2003.

Per quanto riguarda il futuro dei sistemi operativi di Redmond, il nuovo progetto ha come nome in codice “Longhorn”. E' stato più volte rimandato nella data di uscita ufficiale (le ultime comunicazioni dell'azienda rivelano una commercializzazione prevista per giugno 2006) e purtroppo non si dispone ancora di notizie sufficienti a valutare le potenzialità del prodotto.

9 Anche noto come “malicious code”; mancando nella nostra lingua una traduzione soddisfacente, utilizzeremo il termine originale. “Codice malizioso”, infatti, l'espressione più vicina alla versione originale, non riteniamo possa essere adatta.

10 Un *virus* informatico è un programma capace di auto-replicarsi, che si diffonde inserendo copie di se stesso all'interno di codice eseguibile e documenti. Da qui l'analogia con i virus biologici che hanno bisogno di cellule per propagarsi. Più diffusi in passato, sono stati rapidamente sostituiti dagli *worm*; questi ultimi sono caratterizzati dal non dipendere da altri eseguibili o file da infettare al fine di potersi diffondere, essendo autosufficienti ed in grado di replicarsi autonomamente. Sono spesso utilizzati per attacchi alle trasmissioni dati, cancellazione dei file sul sistema infetto o, sempre più frequentemente, invii di email spam. Il worm Mydoom, ad esempio, diffuso nel gennaio 2004, causò il blocco di milioni di computer sfruttando una falla di sicurezza nel codice di Microsoft Windows.

1.2 Le origini di GNU/Linux



Il termine Immagine 1: Il logo di Microsoft Windows.

"Linux" si riferisce sia al nome del sistema operativo che a quello del suo kernel, il cuore di tutto il sistema. Probabilmente rappresenta il più famoso esempio di software libero e del sistema di sviluppo open source¹¹.

Il nome esatto per l'intero sistema operativo, e non solo per il suo kernel, è "GNU/Linux", conseguenza di quello che è stato lo sviluppo del progetto.

Ciò nonostante, il termine usato più frequentemente è proprio Linux, nome di un progetto nato nel 1991 per opera di Linus Torvalds. Questi, ancora studente all'Università di Helsinki in Finlandia, scrisse un kernel di sistema operativo per hobby. Insoddisfatto del sistema operativo usato a scopo didattico durante i corsi, Minix¹², decise di scrivere ex novo un kernel *Unix like*¹³. Il codice ispirato a Minix, per quanto non usasse nessuna parte di codice dell'originale (nelle parole di Torvalds, "*a free version of a minix-lookalike for AT-386 computers*"), fu pubblicato per la prima volta nel 1991

11 Per una migliore comprensione di "software libero" e software "open source", si rimanda al paragrafo successivo.

12 Sistema operativo *Unix like* sviluppato da Andrew S. Tanenbaum, professore di Architettura e Implementazione di Sistemi Operativi, alla Vrije Universiteit, Amsterdam. Nell'idea iniziale del suo sviluppatore, Minix nasceva come supporto didattico e non prevedeva estensioni. Fu rapidamente superato dal kernel Linux e fu utilizzato, proprio come Tanenbaum aveva "previsto", solo in ambito accademico per l'insegnamento.

13 Espressione con la quale si indicano i sistemi operativi derivati da Unix.

su Internet. Fu seguito rapidamente, dopo circa un mese, dalla seconda versione. Da quel momento, quando Linus Torvalds pubblicò il codice sorgente¹⁴ del suo progetto, migliaia di sviluppatori nel mondo parteciparono al progetto, scrivendone parte del codice e contribuendo alla correzione dei bug.

La storia di questo sistema operativo è strettamente legata a *GNU*, un importante progetto di software libero, fondato e guidato da Richard Stallman. Il progetto ebbe inizio nel 1983 al fine di sviluppare un intero sistema operativo *Unix like*, completo di librerie di sviluppo e applicativi, composto esclusivamente da software libero. Il progetto GNU, e la Free Software Foundation nata per sostenerlo, quando nacque Linux, aveva già scritto gran parte del sistema operativo (grazie al contributo di vari sviluppatori), ma mancava del suo componente più importante: il kernel. Questo rappresenta il cuore di qualsiasi sistema operativo e il progetto GNU ancora non era riuscito a scriverne il codice.

L'annuncio del rilascio da parte di Torvalds del kernel Linux fu quindi un'importantissima occasione per il sistema operativo nascente che, finalmente, era completo. Lo studente finlandese, infatti, decise di scrivere il codice in modo da poter funzionare con il software GNU e, soprattutto, di rilasciare il software sotto la licenza *GNU GPL*¹⁵.

14 Per “codice sorgente” di un software si intende la sua “ricetta segreta”, le linee di codice che intimamente ne spiegano il funzionamento. Più avanti procederemo con una più esaustiva descrizione.

15 *GNU General Public Licence*. Licenza ideata e composta dal fondatore del progetto GNU, Richard Stallman. E' considerata da molti l'unica licenza che conceda all'utente i necessari diritti per poter considerare un software *libero*. Nel paragrafo successivo procederemo con una

Dall'unione dei due componenti, il kernel di Torvalds ed il software GNU, si considera “GNU/Linux” il nome corretto con il quale riferirsi al sistema operativo in questione, per dare adeguato risalto anche al lavoro della Free Software Foundation. In realtà, il sistema operativo è noto in tutto il mondo solo con il termine "Linux", aspetto del quale Richard Stallman non manca mai di rammaricarsi¹⁶, e con la sua mascotte, un pinguino di nome Tux.



Immagine 2: Tux il pinguino, simbolo e mascotte del sistema operativo libero GNU/Linux.

1.2.1 Il Free Software

GNU/Linux è forse l'esempio di software libero più importante e di maggior successo. Purtroppo, il significato di “*free software*” è spesso

esposizione dei principi alla base di questo tipo di licenza.

¹⁶ E' abitudine, infatti, che durante tutte le conferenze cui interviene – ad una delle quali ho avuto la fortuna di assistere, nel 2004 presso la Facoltà di Informatica dell'Università di Firenze –, Stallman non tralascia mai di ricordare ai suoi spettatori di usare il termine GNU/Linux. Chiamandolo semplicemente “Linux”, infatti, - spiega Stallman – si manca di rispetto a tutte quelle persone che hanno ideato il software libero, che per anni ci hanno creduto e che lo hanno sviluppato.

frainteso e si scontra con la logica preponderante nell'industria del software e, più in generale, dell'industria in generale.

Scopo di questo paragrafo è proprio quello di fare chiarezza sul concetto di software libero, su cosa implica, su cosa *non* implica e sui diritti che assegna all'utente. La maggior parte di questi concetti sono tratti dalla documentazione, e dalla storia, della Free Software Foundation, fondata da Richard Stallman, “ideatore” del software libero.

Il *free software* è del software che, una volta ottenuto, può essere usato, copiato, studiato, modificato e ridistribuito. Normalmente è di pubblico dominio e non prevede un prezzo da pagare per la licenza. Ciò nonostante uno sviluppatore può vendere software libero, senza ledere minimamente i diritti, sopra menzionati, dell'utente. Uno dei punti che crea confusione nella comprensione di questo concetto è la presenza di un altro tipo di software, il cosiddetto “*freeware*”: questo tipo di licenza (che talvolta rende anche disponibile il codice sorgente) non permette normalmente la modifica del programma e la libera redistribuzione dello stesso. La confusione nasce, oltre che a causa dell'affinità dei termini in inglese, dal fatto che questo tipo di software è gratuito, così come la maggior parte del software libero. Il freeware, però, non può essere in nessun modo considerato *libero*, proprio a causa dei diritti che non assegna in capo all'utente.

Negli anni 60 e 70 del secolo scorso il software non era considerato un prodotto, quanto piuttosto un componente aggiuntivo che i costruttori di

mainframe¹⁷ fornivano ai propri clienti, al fine di poter utilizzare i calcolatori stessi. Tra i programmatori e gli sviluppatori era assolutamente comune la condivisione del codice del software al fine, sia didattico che di ricerca, di migliorarne la qualità e risolverne i problemi. Verso la fine degli anni 70, però, le compagnie sviluppatrici cominciarono ad imporre restrizioni in capo agli utenti del software, tramite la sottoscrizione di particolari licenze d'uso, note come EULA (*End User License Agreement*).

Nel 1984, Richard Stallman, allora ricercatore al MIT, *Massachusetts Institute of Technology*, insoddisfatto del cammino che il mercato del software aveva intrapreso, cominciò a lavorare al Progetto GNU¹⁸, fondando la Free Software Foundation (FSF) un anno dopo. Proprio a Stallman si deve l'introduzione di una definizione di “free software” ed il concetto di “*copyleft*” (con un gioco di parole in lingua inglese ed in evidente contrapposizione con il copyright), ideate al fine di assegnare delle libertà all'utente e di limitare le possibilità di appropriazione del codice.

Stando a questa definizione (tutt'oggi ancora la più completa e citata), il software è libero (*free*) se garantisce:

- la libertà di eseguire, ed usare, il programma per qualsiasi scopo (la

17 Calcolatori usati in quegli anni, celebri per le loro importanti dimensioni. Con il tempo furono sostituiti dai personal computer, che acquisivano sempre maggiore potenza di calcolo a fronte di un costo minore, oltre ad una maggiore facilità di utilizzo e manutenzione. Attualmente vengono utilizzati solo nelle strutture in cui vi è necessità di enorme potenza di calcolo, come strutture militari, centri di calcolo di banche o di grandi industrie, per lo svolgimento di compiti impensabili con altri tipi di calcolatori.

18 GNU, acronimo di “*GNU's Not Unix*”, ovvero “GNU non è Unix”. Questo acronimo è di tipo ricorsivo, usando una tecnica comune nell'ambiente *hacker*, cui Stallman, e la maggior parte degli sviluppatori dell'epoca, appartiene.

cosiddetta “*freedom 0*”)

- la libertà di studiare e modificare il programma (“*freedom 1*”)
- la libertà di copiare il programma “in modo da aiutare il tuo vicino”
[sic] (“*freedom 2*”)
- la libertà di migliorare il programma, e rilasciare pubblicamente i miglioramenti, “in modo che l'intera comunità ne tragga beneficio”
[sic] (“*freedom 3*”)

Da notare come le libertà numero 1 e 3 richiedano come presupposto l'accesso al *codice sorgente*.

Questo è costituito da una serie di istruzioni scritte in un linguaggio di programmazione, composte dallo sviluppatore del software. Rappresenta l'insieme di operazioni che indica al calcolatore cosa deve eseguire per giungere ad un certo risultato. I programmi di tipo proprietario non vengono rilasciati sotto forma di codice sorgente, ma solo in forma binaria. Per giungere a questo stadio, vengono eseguiti i programmi cosiddetti *compilatori* sul codice sorgente stesso; un programma disponibile solo in forma binaria non permette la comprensione dei passaggi che il calcolatore fa, ma mostra esclusivamente l'output prodotto.

Facendo un paragone con il mondo culinario, un programma in forma binaria è assimilabile al piatto pronto, servito e pronto per la degustazione. Sarà possibile apprezzare la bontà della pietanza, ma dalla degustazione non

si potrà mai risalire alla ricetta, ovvero all'esatta sequenza di operazioni eseguite dallo chef per la preparazione del piatto. Analogamente con il software in forma binaria, l'utente-assaggiatore non ha la possibilità di comprendere come quel programma sia stato scritto, ma può solo osservarne il funzionamento ed apprezzarne (o criticarne) la qualità. Allo stesso modo, se lo sviluppatore autore del programma vorrà nascondere all'utente finale certi tipi di operazioni eseguite dal software, egli potrà farlo senza troppa difficoltà; sarà sufficiente non distribuire il codice sorgente del proprio prodotto, ma rilasciarne solo la versione binaria.

Tornando al tema delle libertà concesse ad un utente di software libero, è chiaro come il software proprietario non possa garantire tutte queste facoltà, in particolare quella di poter studiare e modificare il codice e quella di migliorarlo e ripubblicarlo: il concetto di copyright, infatti, assegna solo al proprietario la maggior parte dei diritti di modifica, duplicazione e redistribuzione.

La stessa licenza, poi, impone un altro importante comportamento dello sviluppatore di software GPL: se questi modifica il codice sorgente di un programma di questo tipo, egli sarà tenuto a pubblicarne la versione modificata con la stessa licenza¹⁹. Questo aspetto, insieme ad altri “oneri

¹⁹ A riguardo, considero degna di note la prima sentenza di un tribunale in difesa della licenza GPL. Un tribunale tedesco, infatti, il 14 aprile 2004 ha chiarito che “software libero non è sinonimo di software senza diritto”. Il caso era nato a causa di un'azienda costruttrice di router che aveva usato parte del codice sorgente di *iptables*, applicativo “*firewall*” usato in qualsiasi distribuzione GNU/Linux, rilasciato con licenza GPL. L'azienda, però, violava la licenza in più punti: non accludeva con il prodotto l'intera licenza GPL, non rilasciava con lo stesso tipo di licenza la propria versione modificata e non comunicava agli sviluppatori la modifica stessa. Il tribunale, in questione, ha condannato l'azienda per la violazione della licenza.

minori”, è atto a garantire che nessuno possa appropriarsi del codice sorgente di un software libero, modificarlo e non condividere con altri interessati le modifiche ad esso apportate.

Come accennato precedentemente, la definizione della Free Software Foundation non affronta il tema del prezzo di vendita di un software libero; per meglio spiegare il punto, è molto comune fare ricorso allo slogan “*free as in speech, not as in beer*”. La confusione tra “libero” e “gratuito” è in realtà un problema principalmente legato alla lingua inglese e proprio a questo scopo viene utilizzata questa espressione. In entrambi i casi, infatti, il termine corretto è “*free*”, cui, però, sono legati due significati: “libero” e “gratuito”. Lo slogan ha lo scopo di correlare il software libero alla libertà di parola (“*free speech*”) e non a ciò che può essere gratuito (“*free beer*”).

1.2.2 Free Software e software Open Source: le ambiguità

Un'ulteriore ambiguità è quella creata dall'utilizzo, come sinonimi, delle espressioni “free software” e “open source software”.

Il movimento Open Source nacque circa 15 anni più tardi la fondazione della Free Software Foundation di Stallman, ad opera di un gruppo di sviluppatori, con a capo Eric S. Raymond²⁰ e Bruce Perens, gettando le basi

²⁰ Autore de “*The Cathedral and the Bazaar*”, libro di riferimento per la comprensione del pensiero alla base del movimento open source. Convinto sostenitore di questo testo è anche Marc Andreessen, presidente di Netscape; per sua stessa ammissione, dopo averne letto il contenuto, decise di rivoluzionare l'assetto della propria azienda. La sua software house sviluppava Netscape Navigator, un browser web. Questo prodotto, software di tipo commerciale, leader nel suo mercato nei primi anni 90, presto dovette scontrarsi con Internet

dell'OSI, *Open Source Initiative*.

Il pensiero alla base di questo progetto è quello di rendere maggiori i vantaggi, aumentandone il profilo, della condivisione del codice sorgente del software e di stimolare, ed attirare, l'interesse delle maggiori software house e di altre grandi aziende del settore. A differenza della Free Software Foundation (o meglio, del suo fondatore Richard Stallman), l'OSI non considera l'utilizzo di software open source come un “dovere morale” o un “obbligo nei confronti della società”, ma vede in esso solamente la tipologia di sviluppo software *più adatta ed efficiente*.

L'attenzione principale è rivolta all'eccellenza qualitativa nello sviluppo del software e non ad esigenze etiche o morali; al contrario, Stallman pone il focus maggiore sull'eticità del software libero. Per questo motivo, l'utilizzo come sinonimi dei termini “free software” ed “open source” viene considerato da molti non corretto perché non tiene in adeguata considerazione quali sono le motivazioni alla base dello sviluppo ed all'utilizzo di tale tipo di software.

Ciò nonostante, riteniamo che, avendo chiara la distinzione sopra menzionata, non sia *scorretto* fare uso di entrambi i termini come sinonimi.

Il software in sé, infatti, che sia free o che sia open source è indifferente;

Explorer di Microsoft. Alla fine vinse il prodotto di Redmond, seppur con alterni momenti e con discusse vicende legali a causa del presunto abuso di posizione dominante da parte di Microsoft e che portò l'azienda di Redmond ad un risarcimento record di 750 milioni di dollari. Andreessen, comunque, decise nel gennaio 1998, quando ormai la cosiddetta “guerra dei browser” era già terminata proclamando Internet Explorer vincitore, di rilasciare il codice sorgente di Netscape Navigator, dando inizio al progetto open source Mozilla. Da quel momento, centinaia di programmatori collaborarono alla stesura di Mozilla Browser e, in tempi recenti, a Mozilla Firefox. Di questi prodotti, e della cosiddetta “nuova guerra dei browser”, parleremo più avanti nel corso del lavoro.

nella maggior parte dei casi, infatti, vengono rilasciati tutti con la licenza GPL, sopra menzionata.

1.3 GNU/Linux, un mondo a se stante

Tradizionalmente i sistemi operativi Unix like sono sempre stati destinati ad un'utenza professionale e ad un utilizzo su macchine server. Gli stessi sviluppatori di questo tipo di software non si sono mai preoccupati di rivolgersi ad utenti non esperti, rendendo gli applicativi facili da usare o configurare.

Linux non smentiva questa regola: la sua stessa nascita ed evoluzione è dovuta all'interesse, alla passione ed alla condivisione del codice scritto da migliaia di programmatori: l'utilizzo del sistema operativo e dei programmi ad esso dedicati era destinato ad esperti, ricercatori in ambito universitario abituati a lavorare in ambiente Unix o semplici *hacker*²¹ con il desiderio di capire come effettivamente funzionasse un computer.

La struttura e le logiche di funzionamento del sistema operativo sono state influenzate pesantemente da questo *target* di utenti: questi sistemi, infatti, sono stati progettati con l'obiettivo principale della stabilità e della

21 Mi riferisco all'accezione *tecnica* del termine in questione e non a quella che sempre più spesso si incontra sulla stampa. “*Hacker*”, in questo senso, indica semplicemente una persona appassionata, tendenzialmente esperta, con una infantile curiosità di andare oltre lo strato superficiale del semplice funzionamento di un software, o hardware, e comprenderne gli aspetti più complessi. Ad individui del genere si deve lo sviluppo dei sistemi operativi stessi o, in maniera ancora più netta, della stessa Internet. Sempre più spesso, invece, si confonde “*hacker*” con “*cracker*”, soggetti che utilizzano la propria preparazione per trarne personale guadagno, come nel caso delle truffe telematiche.

sicurezza del sistema. Per il primo aspetto, il codice era scritto al fine di poter reggere grandi carichi di lavoro con continuità e senza perdite sensibili nelle performance, per poter vantare un *uptime*²² di diversi mesi o anni, per poter supportare l'esecuzione in multitasking di diversi applicativi e così via.

Allo stesso modo, gli sviluppatori si preoccupavano della sicurezza, scrivendo software che potesse essere rapidamente aggiornato in caso di scoperta di *bug*²³ gravi, che potesse essere utilizzato effettivamente in multiutenza²⁴ e che, soprattutto, fosse testato da un numero tale di sviluppatori, in possesso del codice sorgente, da poter scongiurare gli errori più gravi o potenzialmente pericolosi e che garantisse un intervento rapido per correggere dette falle. L'essere semplice da utilizzare, o addirittura banale, non costituiva un traguardo da raggiungere né tanto meno da considerare nello sviluppo del software. Le caratteristiche che ora possiamo

22 Con questo termine ci si riferisce al tempo passato dall'ultimo riavvio della macchina. Praticamente ininfluenza in ambito desktop, l'esigenza di un lungo uptime è fondamentale per i server, al fine di non interrompere mai, o soli in rarissimi casi, il servizio prestato (ad esempio, nel caso di un sito web): tale indice viene spesso utilizzato per valutare la stabilità di un sistema operativo o di un applicativo.

23 Letteralmente “insetto”, si intende con questo termine un errore di programmazione nel codice di un software. Evenienza praticamente inevitabile nello sviluppo di un qualsiasi progetto, la presenza di un bug nel codice può avere conseguenze diverse sul funzionamento del software stesso: dall'impossibilità di usare una particolare funzione, ad esempio, al blocco improvviso dell'applicazione stessa o alla creazione di una falla di sicurezza. Quest'ultimo caso è quello più pericoloso in quanto detta falla viene spesso utilizzata dagli sviluppatori di virus e worm per colpire i computer e potersi diffondere. Code Red Worm e SQL Slammer, ad esempio, come illustreremo più avanti, sfruttavano entrambi falle di sicurezza nel codice di, rispettivamente, Microsoft IIS web server e Microsoft SQL Server.

24 Con questo concetto si indica la possibilità di usare un sistema con diversi *account* utente, ognuno caratterizzato da specifiche personalizzazioni e, soprattutto, autorizzazioni a compiere determinate operazioni. Questo costituisce uno dei metodi più efficienti per la protezione di un sistema: l'utente normale, infatti, non può compiere operazioni potenzialmente distruttive, di norma concesse solo all'utente amministratore. In questo modo l'errore umano di un utente non amministratore non può mettere a repentaglio il funzionamento dell'intero sistema, ma solo dell'ambiente proprio dell'utente stesso.

osservare in un sistema operativo come Linux sono frutto proprio di questa mentalità e hanno costituito allo stesso tempo per anni la barriera principale alla diffusione dello stesso.

Negli anni in cui la sicurezza informatica e la stabilità di un software non hanno mai rivestito una particolare importanza, soprattutto per l'utenza desktop, i vantaggi offerti dall'utilizzo di Linux erano assolutamente insignificanti rispetto alla preparazione necessaria per usarlo ed alle sue intrinseche difficoltà e difetti. Questa è una delle ragioni principali che, a nostro avviso, hanno rallentato la diffusione in ambito desktop; l'unico campo in cui Linux poteva giocare un ruolo di rilievo era quello dei server, anche se costantemente in ritardo rispetto ad altri prodotti.

In questo ambito i concorrenti più temibili erano i diversi sistemi Unix proprietari, come *Solaris* di *Sun*, molto diffusi in ambito universitario e di ricerca, o gli stessi prodotti server di Microsoft, presenti principalmente in ambito aziendale. Linux, come software libero e come tale modificabile ed adattabile da chiunque ne avesse le capacità, mostrava di avere grandi potenzialità che solo diversi anni di sviluppo avrebbero potuto esprimere e sfruttare al meglio.

Il progetto, infatti, era troppo giovane e poco maturo per potersi diffondere in ambiti così esigenti come quelli di ricerca o di grandi centri di elaborazione dati. Linux doveva ancora dimostrare di essere in grado di superare, in quanto a stabilità, performance e sicurezza, i prodotti che fino a

quel momento dominavano il mercato. Lo avrebbe dimostrato, ma era ancora troppo presto. Linux è rimasto così appannaggio di pochi esperti e sperimentatori curiosi con la voglia di esplorare un nuovo modo di usare il personal computer, a confronto con quella che era la soluzione più diffusa, Microsoft Windows.

1.4 Windows: un'ascesa inarrestabile

Le motivazioni alla base di una così rapida ed inarrestabile diffusione di Windows sono, a nostro parere, molteplici: il sistema operativo di Microsoft, infatti, ha conosciuto un'evoluzione piuttosto controversa, come illustrato nel primo paragrafo.

Le qualità del prodotto, una conduzione d'azienda vincente, una strategia commerciale agguerrita (che, come visto, più volte ha suscitato l'interesse delle corti dei tribunali), l'illuminante anticipazione dell'andamento del mercato formulata dal leader Bill Gates, hanno reso Microsoft Windows il sistema operativo installato su oltre il 90% dei personal computer nel mondo. Sicuramente, la strategia di sviluppare un prodotto semplice, facile da usare, piuttosto che qualitativamente eccelso, ha prodotto risultati entusiasmanti e, soprattutto, ha reso possibile un'informatizzazione globale che, forse, non sarebbe avvenuta con le altre soluzioni software diffuse in quegli anni.

La scelta di non sviluppare un sistema hardware proprietario, a differenza di

quanto fatto da Apple, ad esempio, ha reso Windows una piattaforma vincente e disponibile per la maggior parte del parco macchine esistente.

Microsoft, inoltre, si è mostrata capace di un'efficiente strategia commerciale, stringendo accordi con i costruttori di hardware al fine di *preinstallare*²⁵ Windows sui propri prodotti; in un periodo nel quale non si era ancora affermato definitivamente un solo sistema operativo e quando il fenomeno dei personal computer era ancora in continua evoluzione, questo comportò un aumento netto delle installazioni di Windows, a discapito degli altri produttori software.

L'azienda di Redmond, inoltre, si mostrò anche molto lungimirante in merito a cosa il futuro avrebbe riservato al fenomeno dei personal computer: l'intrattenimento multimediale. Sin dall'uscita nel 1990 di “Windows 3.0 with Multimedia Extension”, corredato di scheda audio e lettore CD-Rom, Microsoft dimostrò di voler scommettere sull'utenza desktop e di intrattenimento. L'azienda, così, strinse accordi con i più importanti produttori di periferiche per personal computer, garantendo la compatibilità pressoché totale del proprio sistema operativo con l'hardware esistente.

Come già anticipato, inoltre, la vera killer application di Windows è stata rappresentata da Microsoft Office, la suite di produttività personale più

²⁵ Per quanto sia ormai prassi comune, la preinstallazione del sistema operativo su un personal computer non è sempre stata presente. Una strategia di questo tipo comporta per il cliente un prezzo di acquisto dell'hardware comprendente anche il costo della licenza del proprietario del software preinstallato. Uno dei vantaggi principali, però, è di poter acquistare un personal computer già configurato e pronto all'uso.

diffusa al mondo. Da sempre considerato un ottimo prodotto, sin dalle primissime versioni l'utenza aziendale, piuttosto che quella semplicemente domestica ancora non partecipe della diffusione dei personal computer, aveva dimostrato un notevole interesse e fece vendere il prodotto in milioni di copie. L'effetto del binomio Windows – Office non tardò di mostrarsi: le vendite di Office andavano di pari passo con quelle del sistema operativo, garantendo una diffusione senza precedenti.

Siamo inoltre dell'opinione che le operazioni “controverse” compiute da Microsoft nel corso del tempo, come sentenze di tribunale e accordi pregiudiziali di risarcimento sembrano indicare, non abbiano giocato un ruolo determinante nella diffusione dei prodotti dell'azienda statunitense. E' certamente possibile che Internet Explorer non avrebbe conquistato l'intero mercato nel volgere di pochi mesi se Microsoft non l'avesse integrato nel sistema operativo, ma l'obiettivo sarebbe stato comunque raggiunto, seppur in un tempo maggiore. Windows, inoltre, era l'unica alternativa alla piattaforma Apple, da sempre elitaria e caratterizzata da un prezzo elevato, e ad OS/2 di IBM, poco diffuso e carente delle applicazioni più importanti. Nel mercato desktop, quindi, Windows non aveva di fatto concorrenti diretti in grado di metterne a repentaglio la leadership; più complessa, invece, la sua posizione in ambito server.

Gli sforzi di Microsoft verso questo mercato si sono concretizzati, in un primo momento, in Windows NT che, però, conobbe all'inizio una certa

diffidenza che portò ad uno scarso successo del progetto a tutto vantaggio della versione “desktop” di Windows, aggiornato con un supporto della gestione di rete migliorato, come nel caso di “*Internet Connection Sharing*” di Windows 98.

In tempi più recenti i prodotti Microsoft più apprezzati ed utilizzati in ambito server sono rappresentati dall'ottimo Windows 2000, la cui qualità e bontà del codice è riconosciuta anche dai più acerrimi detrattori dell'azienda produttrice, Windows XP Professional e, recentemente, da Windows 2003 Server.

2. Perché migrare

In questo capitolo è nostra intenzione illustrare le motivazioni che rendono Linux ed il software open source una realtà concreta per le aziende ed una possibilità di ottenere vantaggi importanti.

Le motivazioni che possono spingere l'amministrazione a migrare verso sistemi Linux, abbandonando un'infrastruttura basata su prodotti Microsoft, sono a nostro avviso diverse. Le motivazioni principali che il nostro lavoro affronterà sono:

- Sicurezza
- Flessibilità
- Facilità di amministrazione
- Costi
- *Vendor tie-in*

Di seguito, le illustreremo singolarmente.

2.1 Il problema della Sicurezza Informatica: panoramica

In questi ultimi anni il mondo dell'Information Technology ha conosciuto importanti cambiamenti che, a nostro avviso, hanno modificato

profondamente lo scenario e che hanno costretto, o costringeranno, gli utenti a cambiare radicalmente le proprie abitudini e l'approccio nei confronti della tecnologia. La rapida ed incessante diffusione di Internet e del conseguente numero di computer connessi, ha mostrato al mondo uno dei suoi aspetti più preoccupanti: la sicurezza informatica.

In questo paragrafo è nostra intenzione descrivere i casi più importanti di “attacchi informatici” e i danni da essi provocati per illustrare meglio lo scenario relativo e le ripercussioni tecniche ed economiche che provoca.

26 marzo 1999: nelle prime ore del mattino viene individuato uno dei primi casi di infezione del worm *Melissa*¹ negli Stati Uniti. Nel volgere di poche ore il “*malware*” si diffonde in tutto il globo con una velocità fino a quel momento mai registrata per un software del genere. Nelle ore serali vengono infettati i primi personal computer in Europa; Microsoft è costretta ad interrompere integralmente il proprio servizio di posta elettronica per contenere i danni. La stessa azienda, così come la *Intel Corporation* ed altre grandi multinazionali, riportano che l'infezione ha colpito la quasi totalità della propria infrastruttura; il numero di computer colpiti da questo worm è stimato in decine di migliaia ed è in rapido aumento. Le dichiarazioni di un esperto in materia, Mikko Hypponen, responsabile di *F-Secure* per la divisione “*Anti-virus Research*”, sono chiare: non era mai stato visto un worm del genere e dalle potenzialità così pericolose².

1 Noto anche sotto il nome di: *Simpsons, Kwyjibo, Kwejeebo, Mailissa, W97M*.

2 Riportiamo in inglese un estratto delle comunicazioni ufficiali: “We've never seen a virus spread so rapidly. We've seen a handful of viruses that distribute themselves automatically over e-mail, but not a single one of them has been as successful as Melissa in the real world.”. Cfr.

Il codice si diffuse inizialmente tramite un forum, mostrando un file allegato dal nome “*List.doc*”, almeno nella sua versione originaria. Visualizzando l'allegato, il codice veniva attivato e Melissa cominciava ad infettare tutti i file per Microsoft Word presenti sul sistema ospite. Il funzionamento del worm proseguiva, e questo fu l'effetto più dannoso dell'infezione, con l'invio tramite posta elettronica dei file *doc* per Microsoft Word infetti, rischiando, in questo modo, che venissero trasmesse anche informazioni confidenziali. Naturalmente, il codice del worm non consentiva all'utente del computer di accorgersi del massiccio invio di email che avveniva in background e così, in brevissimo tempo, interi sistemi per il recapito della posta elettronica furono completamente bloccati. Il codice era in grado di funzionare, ed infettare quindi il sistema ospite, solo qualora venisse attivato tramite Outlook 97 e Outlook 98, il software di posta elettronica compreso nella suite Office di Microsoft, e diffuso tramite Word. Melissa è considerato il primo worm³ ad aver causato una ingente quantità di danni nel volgere di un lasso di tempo così breve.

Circa un anno dopo, nel maggio del 2000, un altro worm entra nella storia per aver causato i maggiori danni economici: *Loveletter Worm*⁴ entra in azione.

Il codice era scritto in *Visual Basic Script (VBS)*, da qui uno dei suoi nomi),

<http://www.f-secure.com/v-descs/melissa.shtml>

3 Nella storia di questo particolare tipo di programmi, il capostipite è “*Elk Cloner*”, scritto per il sistema Apple II. E' accreditato di essere il primo ad essere comparso “*in the wild*”, ovvero all'esterno del computer o della struttura in cui era stato creato. Questo comportamento, come visto, è ormai la norma per i moderni virus e worm.

4 Noto anche come *VBS*, *Iloveyou* e *Lovebug*.

linguaggio di programmazione creato da Microsoft, derivato dal Basic e utilizzato anche per *Asp*⁵ (*Active Server Pages*) nei software per server della stessa azienda.

Le prime infezioni sono state rilevate ad Hong Kong e, anche in questo caso, il worm si trasmetteva tramite allegato di posta elettronica. Una volta colpito il sistema ospite, il software infettava la maggior parte dei file presenti sul computer e si auto-inviava per email tramite Microsoft Outlook a tutti i contatti presenti nella rubrica dello stesso programma. Il programma in questione, inoltre, era stato scritto per creare delle *backdoor*⁶ all'interno dei sistemi infetti, al fine di usarli per attaccare, tramite un *Denial Of Service*⁷, il sito della Casa Bianca americana. Il danno derivante dall'infezione di questo worm è stato stimato in otto miliardi di dollari.

Nei due esempi di infezioni qui riportati il fattore umano si è dimostrato determinante; per quanto il codice di *Melissa* e di *LoveLetter* sfruttasse dei software installati sulla macchina ospite (Outlook e Word), ciò avveniva in una fase successiva dell'infezione. Il momento determinante, infatti, consisteva nell'attivazione del “*malware*”, tramite la lettura dell'allegato di posta elettronica. In questo caso un attento utilizzo dello strumento e una

5 Tecnologia *server-side* di Microsoft per la creazione di pagine web dinamiche. Scritto per essere eseguito da *IIS*, *Internet Information Server*, server web della stessa azienda.

6 Letteralmente “porta sul retro”, in ambito informatico consiste nella creazione di “una testa di ponte” sul sistema colpito in modo tale da poterne assumere, successivamente, il controllo da remoto.

7 *DoS*, o più correttamente in questo caso *DDoS*, *Distributed Denial of Service*. Attacco ad un sistema informatico consistente nell'eseguire contemporaneamente da più macchine una richiesta allo stesso server. Quest'ultimo, non riuscendo a sopportare il carico di tutte le connessioni, “cade” e non è più in grado di rispondere, rendendosi inaccessibile, finché le richieste non cessino.

buona dote di lungimiranza permetteva di passare indenne alla diffusione dei due worm in questione⁸.

Ciò che avvenne nel luglio dell'anno successivo, invece, fu completamente diverso: il 13 del mese il worm "*Code Red*" colpì, indistintamente, tutti i server web che utilizzavano IIS, il software di *web server* Microsoft. Questa volta l'autore del programma non sperava nell'ingenuità di qualche utente per propagarsi, ma decise di sfruttare una falla, un bug del server web di Microsoft.

La vulnerabilità in questione era relativa al codice interno di IIS per l'indicizzazione dei dati e Code Red ne eseguiva l'*exploit*⁹. Gli effetti erano vari: il sito veniva “defacciato” (*defaced*), ovvero sostituito con una pagina composta dall'autore del worm; la macchina infetta veniva usata per cercare altri server web IIS in Internet; dopo circa un mese tutte le macchine infette eseguivano un *DDoS* verso il sito web della Casa Bianca, per la seconda volta in pochi mesi, che diventò irraggiungibile nel giro di poche ore.

Il bug sfruttato da Code Red era noto da circa un mese¹⁰ e qualche settimana prima era stata pubblicata la *patch*¹¹ necessaria a “tappare la falla”. La

stessa Microsoft fu colpita dall'infezione, dimostrando che i propri sistemi,

8 Kevin Mitnick, forse il più celebre *hacker* della storia, utilizzava per questo l'espressione “ingegneria sociale”: spesso – sostiene lo statunitense – la tecnica più semplice per violare un sistema informatico è quello di attaccare il “fattore umano”. Indovinare le semplici password utilizzate dalla maggior parte degli utenti o inoltrare un allegato di posta dal nome interessante, può essere sufficiente per penetrare un sistema. Cfr. Mitnick, K.D., *L'arte dell'inganno*, Milano, Feltrinelli, 2003.

9 Tecnica di attacco sfruttando la vulnerabilità, o *bug*, di un software.

10 La prima comunicazione della vulnerabilità, ad opera di eEye Digital Security, è del 17 luglio 2001. Cfr. <http://www.eeye.com/html/Research/Advisories/AL20010717.html>.

11 Letteralmente “toppa”, rappresenta un software aggiuntivo da eseguire su di un programma già installato. Usato anche per estendere le funzionalità di un prodotto, viene frequentemente utilizzato per risolvere bug nel codice.

così come tutti gli altri attaccati dal worm, non erano aggiornati con le ultime patch.

Due mesi dopo, nel settembre del 2001, si scatenò un'altra infezione dagli effetti devastanti sulle infrastrutture di rete: si diffuse il worm "*Nimda*".

Questa volta si assistette al primo caso di malware che infettasse server web al fine di modificarne il contenuto. Nimda sfruttava, anche in questo caso, una vulnerabilità di IIS, il noto bug *Unicode*. Una volta colpito il server, ne modificava il contenuto delle pagine, inserendo al loro interno collegamenti al codice del worm stesso. In questo modo, navigando il contenuto di uno dei siti infetti, un qualsiasi utente Windows¹² avrebbe letto il codice in questione, infettando la propria macchina. Nimda, inoltre, utilizzava anche i computer desktop colpiti al fine di ricercare nella rete nuovi server web IIS da infettare. Nimda riuscì a paralizzare intere strutture informatiche, ivi comprese quelle logistiche di trasmissione dati, con il risultato che intere infrastrutture risultarono tagliate fuori dalle connessioni.

Dopo qualche settimana, in ottobre, fu la volta di "*Klez*", un altro worm che non ebbe bisogno dell'errore umano per colpire e propagarsi. Sfruttando una vulnerabilità di Internet Explorer, presente anche in Outlook e Outlook Express nel codice utilizzato per visualizzare le email in *HTML*¹³. In questo caso l'anteprima di un messaggio di posta elettronica, funzione

12 Il codice funzionava per le seguenti versioni, praticamente tutte, di Windows: 95, 98, Me, NT 4, 2000.

13 *HyperText Markup Language*. Linguaggio per scrivere ipertesti, è rapidamente diventato lo standard dei documenti web. Nonostante sia sconsigliato, per ragioni di sicurezza e di prestazioni, l'utilizzo di questo linguaggio per le semplici email, si è rapidamente diffuso anche per questo strumento.

normalmente abilitata sulla maggior parte dei client di posta elettronica, era sufficiente per l'esecuzione del codice del worm e per la sua propagazione.

La rassegna prosegue con altre due importanti infezioni, entrambe risalenti al 2003, che provocarono ingenti danni: gli worm "*Sobig*" e "*Blaster*".

Il primo, diffusosi principalmente nell'agosto del 2003, colpì milioni di computer.

Modificato in un largo numero di varianti, più o meno dannose, è famoso soprattutto per la sua ultima versione, *Sobig.F*, che raggiunse il record per il volume di email da esso generate. In questo caso, il codice era di maggior qualità: *Sobig* era un worm, e come tale in grado di duplicarsi da solo, ma allo stesso tempo era un "cavallo di Troia" ("*Trojan horse*") perché riusciva a mascherarsi, facendo ritenere al sistema operativo ed ai relativi anti-virus di essere un programma comune. Apparve la prima volta come allegato di posta elettronica e una volta installatosi sul sistema ospite, cominciava la sua attività: eseguiva un software *SMTP*¹⁴ per l'invio della posta verso i contatti contenuti nella rubrica, creava una backdoor in grado di aggiornare il codice del worm automaticamente collegandosi ad una risorsa *online*, infettava inoltre un largo numero di file presenti sul computer colpito.

Il worm "*Blaster*"¹⁵ si diffuse nello stesso periodo di "*Sobig*", durante l'estate 2003. In questo caso i sistemi vulnerabili all'attacco erano i sistemi

¹⁴ *Simple Mail Transfer Protocol*. I server di questo tipo assolvono la funzione di invio della posta elettronica. Per la ricezione delle email, invece, vengono utilizzati i server *POP* o *IMAP*.

¹⁵ Noto anche come *Lovsan* o *Lovesan*.

più recenti di Microsoft, principalmente Windows XP e Windows 2000. Nei sistemi NT, Windows XP 64bit e Windows 2003, infatti, il worm causava solo una generale instabilità di sistema, senza riuscire a propagarsi e senza causare danni degni di nota. Il picco nel numero di infezioni fu raggiunto il 13 Agosto (circa 1,4 milioni di sistemi in tutto il pianeta¹⁶), circa due giorni dopo la prima infezione, ad un velocità di circa 2500 nuove infezioni ogni ora. Il worm fu scritto per eseguire un *DDoS* contro il sito web di Microsoft, tramite il quale vengono resi disponibili gli aggiornamenti (compresi quelli di sicurezza) del software; fortunatamente i danni riportati dal sito in questione furono minimi perché, in realtà, questo eseguiva un reindirizzamento (*redirect*) su un altro indirizzo.

L'importanza di questi due ennesimi episodi di diffusione di malware sono rilevanti sia dal punto di vista tecnico che da quello politico: per il primo aspetto, perché conseguirono il record, fino a quel momento, per il maggior *down-time*¹⁷ e per i costi sostenuti per la risoluzione dei danni ai sistemi, mai incontrati fino a quel momento. Lentamente il mondo dell'Information Technology si rese conto, a sue spese, di quanto gravi potessero essere le conseguenze dovute alla compromissione dei sistemi informatici e quanto importante fosse diventato curare seriamente la sicurezza dei network.

Le conseguenze politiche, invece, si concretizzarono in richieste di grandi aziende in diversi paesi rivolte ai rispettivi governi al fine di prevenire

¹⁶ Fonte: *The Computer Emergency Response Team (CERT) Coordination Center*.

¹⁷ Tempo necessario per rendere nuovamente operativo un sistema informatico o una singola macchina.

ulteriori ingenti danni causati da virus e worm per Windows, reclamando una responsabilità del produttore per l'affidabilità dei propri software.

Un comportamento analogo si ebbe in seguito alla diffusione di un altro worm, il cui obiettivo era i server *Microsoft SQL*¹⁸: "*SQL Slammer*". In questo caso la reazione politica fu localizzata principalmente nella sola Corea del Sud, paese che fu letteralmente paralizzato il 25 gennaio 2003 a causa dei danni provocati dal worm in questione ed isolato *completamente* dalla rete Internet. L'*exploit* utilizzato per infettare i server sfruttava due bug, noti, del server SQL di Microsoft. Fortunatamente il numero di infezioni non fu altissimo, ma impressionante ne fu la rapidità: la maggior parte delle 75.000 vittime fu colpita nell'arco di dieci minuti, il 25 gennaio 2003. L'attacco causò un rallentamento significativo del traffico Internet globale. Gli stessi server di Microsoft furono compromessi e furono disattivati per contenere i danni.

L'azienda di Redmond fu inoltre citata in tribunale dal *People's Solidarity for Participatory Democracy*, ONG coreana, insieme a circa tremila privati cittadini, per rispondere delle proprie responsabilità nell'infezione causata dal worm SQL Slammer. Forti della legislazione coreana che rende un fabbricante responsabile dell'affidabilità dei prodotti venduti, la ONG ha richiesto a Microsoft, considerata inadempiente per i suoi obblighi, un importante risarcimento.

¹⁸ *Structured Query Language*, linguaggio usato nel campo dei database, sempre più frequentemente utilizzati anche in ambito web al fine di supportare siti internet dinamici.

Gli ultimi episodi di infezioni diffuse risalgono alla primavera del 2004. Gli artefici sono stati "MyDoom", attuale detentore del record come il worm che ha conosciuto la più rapida diffusione, e "Sasser".

"MyDoom"¹⁹, dalle ricerche fatte, sembra che sia stato commissionato da aziende specializzate in "email spamming"²⁰, al fine di generare un enorme traffico di posta pubblicitaria, utilizzando per l'invio i computer infetti. Purtroppo questa sembra essere diventata una pratica molto diffusa: programmatori si offrono alle aziende per colpire migliaia di sistemi facilmente infettabili, i cosiddetti "zombie PC", utilizzati poi per l'invio di posta indesiderata. Lo stesso codice del programma conteneva indicazioni ("I'm just doing my job, nothing personal, sorry") che avvalorano l'ipotesi dell'infezione su commissione. Una volta aperto l'allegato di posta in cui era contenuto il worm, questi si inviava automaticamente a tutti gli indirizzi presenti nella rubrica e si inseriva anche in apposite *directory* utilizzate dai software di *peer to peer*²¹. E' stato stimato che MyDoom abbia causato un danno economico quantificabile in 40 miliardi di dollari²².

19 Noto anche con gli alias *Novarg*, *Mimail.R* e *Shimgapi*.

20 Uno delle attività illecite più redditizie svolte tramite Internet. Consiste nell'invio di posta indesiderata, normalmente con informazioni pubblicitarie. Il volume di email necessario al fine di ottenere vantaggi concreti da questa attività è enorme; l'analisi del fenomeno, infatti, mostra che solo una minima parte delle email spam ricevute vengono "attivate" tramite click sui link in esse contenute. Attualmente la percentuale del traffico generato da questo tipo di attività sul totale delle email inviate è considerevole e sono allo studio tecnologie in grado di limitarle. Recentemente, il 5 Novembre 2004, lo Stato della Virginia negli USA ha emesso la prima importante e dura sentenza nei confronti di una coppia di *spammer*. Cfr. <http://www.oag.state.va.us>

21 Noti anche come "software di *sharing*", sono di norma utilizzati per il download di file audio, video e di programmi dalla rete, nella maggior parte dei casi violando le leggi sul copyright. Il capostipite fu *Napster*; una volta pubblicato il codice sorgente di questo software, dopo le cause legali intentate contro lo sviluppatore, nacquero numerose varianti di questo programma.

22 Fonte: mi2g, *Report "Damage caused by Malware and Digital Attacks"*, luglio 2004. La società autrice dello studio, mi2g di Londra, è specializzata in sicurezza informatica per grandi aziende e digital risk management.

L'ultima infezione particolarmente interessante è imputabile ad un nuovo worm, "Sasser". Destinato ai recenti sistemi operativi Windows XP e 2000, questo software sfruttava una falla nei prodotti Microsoft in questione e non era necessario alcun intervento umano per attivarsi. L'ingegneria sociale, quindi, non ha giocato nessun ruolo nella diffusione di questo pericoloso malware; era sufficiente un collegamento ad Internet, anche senza visitare siti web o utilizzare la posta elettronica, tramite un personal computer con uno dei due sistemi operativi in questione per essere colpiti *in pochi minuti* e contribuire alla diffusione. Unico rimedio consisteva nell'utilizzo di una patch apposita rilasciata da Microsoft. Il worm, però, una volta colpito il sistema, impediva il collegamento ai siti internet dell'azienda, rendendo impossibile l'aggiornamento di un computer infetto.

Tra i milioni di computer colpiti, si annoverano le strutture informatiche dell'agenzia stampa francese Agence France-Presse che portò al blocco completo delle comunicazioni via satellite dell'agenzia stessa, la compagnia aerea statunitense Delta Air Lines che dovette cancellare diversi voli tra Stati Uniti ed Europa, il gruppo bancario finlandese Sampo Bank che fu costretto a chiudere temporaneamente i 130 uffici del paese. L'elenco delle "illustri vittime" prosegue con la Guardia Costiera britannica che ebbe tutti i servizi di localizzazione geografica bloccati, la Commissione Europea con le proprie infrastrutture informatiche bloccate per ore, Deutsche Post ed, infine, il reparto radiografie dell'ospedale universitario svedese di Lund

(Lund University Hospital) che risultò inutilizzabile per ore, costringendo i pazienti a rivolgersi ad altre strutture.

2.1.1 Chiarezza sul tema della sicurezza informatica

E' necessario fare delle precisazioni sulle conclusioni che si possono trarre dai dati forniti nel paragrafo precedente. Worm e virus sono software; come tali, i rispettivi autori scrivono il codice al fine di poter attaccare un determinato sistema informatico. Quali che siano le motivazioni specifiche che possano spingere un programmatore a creare simili software, l'obiettivo di fondo è quello di poter colpire il maggior numero possibile di sistemi, arrecando danni più o meno ingenti.

La diffusione di Windows ha portato a due conseguenze che giocano un ruolo molto importante in questo aspetto: la sostanziale omogeneità del parco software presente nel mondo e la generale “inesperienza” di parte degli utenti.

Dati i numeri che rappresentano la quota di mercato del sistema operativo Microsoft, con percentuali superiori al 90%, i programmatori di worm sanno di trovarsi dinanzi ad una moltitudine di personal computer che possono essere attaccati con la stessa tecnica e decidono così di colpire il sistema più diffuso.

Oltre a questo aspetto pragmatico, è a nostro parere anche ravvisabile una

certa avversione, da parte degli autori di questi software, nei confronti dell'azienda Microsoft, spesso da questi accusata di sviluppare prodotti scadenti, che può spingere a colpire preferibilmente questi sistemi piuttosto che altri. Questo deve essere, a nostro avviso, un primo strumento per poter meglio collocare nel contesto i dati da noi forniti sugli worm e sui loro “successi”.

Altro aspetto, a nostro parere determinante per il tema in questione, è costituito dalla cultura informatica e dalle conoscenze informatiche, possedute da parte degli utenti Windows, target naturale del prodotto Microsoft. L'azienda di Redmond ha sempre perseguito l'obiettivo di una vasta informatizzazione globale, soprattutto grazie ai propri prodotti; come già affrontato, la comunità di sviluppatori di Linux, al contrario, non ha mai perseguito tali obiettivi, se non nel periodo più recente. Le tematiche legate alla sicurezza informatica, d'altro canto, non sono mai stati il focus più importante di Microsoft. Come spesso accade con i produttori di software proprietari, le esigenze più pressanti nel rilascio di un prodotto sono spesso costituite più dalle nuove funzionalità che dalla stabilità e sicurezza del codice.

E' sufficiente controllare la frequenza con la quale Microsoft rilascia gli aggiornamenti per i propri prodotti, le cosiddette patch di sicurezza, e la lista, costantemente *in fieri*, dei bug scoperti e non ancora risolti. D'altronde è la stessa utenza che non si è mai preoccupata troppo delle gravi

conseguenze correlate ad una incauta configurazione o utilizzo del software, a maggior ragione in tempi, come quelli attuali, in cui il numero di macchine connesse, spesso permanentemente, ad Internet è sempre maggiore.

Linux da questo punto di vista, invece, denuncia una sorta di “conservatività” nella gestione del codice: i rami di sviluppo sono divisi in “*stable*” e “*development*”. Gli inserimenti di nuovo codice nel primo ramo sono definiti da rigide regole di controllo e di *testing* necessario prima di poter essere accolte; il secondo ramo, invece, è destinato principalmente agli sviluppatori ed è sconsigliato negli ambienti in cui sicurezza e stabilità sono delle esigenze fondamentali. L'utenza di riferimento di Linux, poi, è da sempre legato ad una maggiore cultura informatica e ad una maggior attenzione sui temi della sicurezza. *Necessariamente*, dato che fino a qualche anno fa la semplice procedura di installazione di una qualsiasi distribuzione Linux era affrontabile solo da soggetti esperti.

Nel paragrafo 2.1, sono stati illustrati anche i casi in cui la diffusione di un worm è stata frutto di una intelligente “ingegneria sociale” e non dalla perizia del suo autore nella scrittura del codice. Dinanzi a questo tipo di comportamento, incauto e non riflessivo, un *qualsiasi* sistema informatico può subire danni. Proteggere una rete di computer dalle trappole dell'ingegneria sociale è molto difficile, se non tramite una maggiore consapevolezza dei rischi e delle tecniche per difendere al meglio un

sistema, necessariamente correlati, però, ad una maggior cultura informatica dell'utente medio, spesso inesperto, target preferito dagli autori di worm e virus. Muovendo da queste premesse, risulta anche molto complicato proporre soluzioni ai rischi prodotti dalla ingegneria sociale, indissolubilmente legati a barriere culturali e psicologiche di difficile risoluzione. Un adeguato investimento sulla formazione degli utenti porterebbe a validi risultati, almeno nel medio lungo periodo. A maggior ragione diventa sempre più importante fare in modo che l'ingenuità di un utente non provochi danni importanti all'intera rete informatica, bloccando servizi e causando ingenti danni economici. La scelta dell'infrastruttura e una sua oculata configurazione sono fondamentali per la protezione di un sistema, esulando dall'errore umano dovuto ad inesperienza, rischio cui ogni sistema è esposto.

Scopo di questa parte del lavoro non è quello di illustrare come difendersi dall'ingegneria sociale, ma mostrare come Linux possa essere uno strumento per migliorare la sicurezza della propria infrastruttura.

2.1.2 Windows e Linux “*in the wild*”

La nostra analisi della sicurezza dei due sistemi operativi continua con un interessante studio di *Avantgarde*, condotta dal celebre *hacker* Kevin Mitnick e Ryan Russel, e una seconda ricerca svolta da *Honeynet Project*,

progetto indipendente dedicato alla diffusione di contenuti inerenti la sicurezza informatica.

Il primo studio consisteva nel collegare ad Internet, tramite connessione a banda larga, diverse macchine, equipaggiate con diversi sistemi operativi, tra i quali alcune versioni di Windows, una distribuzione di Linux e un computer Apple con MacOS X. Scopo dell'esperimento era di verificare il numero di attacchi che le macchine avrebbero subito nell'arco di due settimane, il numero di “compromissioni”, ovvero di attacchi andati a buon fine, e il tempo necessario affinché un sistema risultasse compromesso.

Importante notare che i sistemi operativi sono stati installati nella configurazione predefinita (di *default*) distribuita dal produttore e che i computer stessi non siano stati mai usati nel periodo del test; questa, infatti, era l'unica possibilità per valutare la sicurezza di un sistema, prescindendo da incauti comportamenti umani o particolari configurazioni da “esperti”, eliminando quindi le distorsioni che l'ingegneria sociale avrebbe comportato.

Analogamente, per rendere il parco macchine dell'esperimento più simile a quanto incontrato comunemente sui computer desktop, sono state impostate password di accesso banali, il più delle volte basate su parole comprese in un dizionario²³.

²³ La precisazione è resa necessaria dall'esistenza di due tecniche principali per il “*password cracking*”, ovvero per gli attacchi compiuti tramite tentativi di indovinare le parole chiave. Un metodo, *brute force*, consiste nell'utilizzare software appositi e la potenza di calcolo di un elaboratore al fine di provare tutte le possibili combinazioni. In questo caso, la scelta di una password composta da numerosi caratteri comporterebbe un tempo di elaborazione necessario troppo lungo per essere colpiti. Un'altra tecnica, invece, utilizza dizionari per “indovinare” la

Il totale degli attacchi subiti dai diversi sistemi ammontava a più di trecentomila (305.955), con un risultato che probabilmente stupirà molte persone: gli attacchi informatici non sono destinati solo a server di banche o di grandi aziende, ma possono colpire *indiscriminatamente* tutti. Dall'analisi condotta sui tentativi in questione, infatti, è emerso come siano tutti originati da software appositamente scritti per cercare in Rete computer collegati, valutarne le difese e colpire. Non è in alcun modo collegato alla scelta dell'obiettivo da compromettere.

I sistemi Windows analizzati, ed installati sulle macchine di prova, erano: Windows XP Service Pack 1, Windows XP Service Pack 2 e Windows Server 2003. Il primo ha subito il 45,44% degli attacchi risultando compromesso per nove volte; la prima infezione si è verificata *in quattro minuti*. La versione aggiornata a Service Pack 2, invece, ha subito lo 0,45% degli attacchi resistendo senza difficoltà a tutti. Dati meno confortanti per il prodotto di fascia server di Microsoft, Windows Server 2003, che, pur subendo solo l' 8,24% degli attacchi, è risultato compromesso in meno di otto ore.

I sistemi non Windows, invece, erano rappresentati da una macchina Linux e una MacOS X. Il sistema Linux non è mai stato compromesso ed ha subito lo 0,26% degli attacchi. Analogo risultato è stato conseguito da MacOS X, resistendo a tutti gli attacchi subiti, pari al 45,32% del totale,

password, data l'incauta abitudine di molti utenti di scegliere parole semplici come chiavi d'accesso; in questo caso, ovviamente, il tempo necessario a trovare la combinazione corretta è di gran lunga minore.

nell'arco delle due settimane dell'esperimento..

<i>Totale attacchi per sistema operativo</i>	<i>Attacchi</i>	<i>% sul totale degli attacchi</i>
Windows Server 2003	25222	8.24%
Windows XP SP1	139024	45.44%
Windows XP SP1 con firewall software	848	0.28%
Windows XP SP2	1386	0.45%
MacOS X 10.3.5	138647	45.32%
LinSpire Linux	795	0.26%

Tabella 1: Risultati dell'esperimento di Avantgarde. Distinzione per numero di attacchi subiti. Fonte: Mitnick, K., Time to Live on the Network, San Francisco, Avantgarde, 2004.

<i>Totale compromissioni per sistema operativo</i>	<i>Compromissioni</i>	<i>Tempo necessario per la compromissione</i>
Windows Server 2003	1	meno di 8 ore
Windows XP SP1	9	meno di 4 minuti
Windows XP SP1 con firewall software	0	0
Windows XP SP2	0	0
MacOS X 10.3.5	0	0
LinSpire Linux	0	0

Tabella 2: Risultati dell'esperimento di Avantgarde. Distinzione per numero di compromissioni subite dai diversi sistemi operativi e tempo necessario per l'infezione. Fonte: op.cit.

Studiando la procedura seguite per condurre l'esperimento, è a nostro avviso doveroso fare una precisazione inerente il caso Windows XP SP 2. Nell'aggiornamento in questione, infatti, Microsoft ha inserito importanti funzionalità nel proprio sistema operativo, precedentemente non presenti (e, quindi, assenti nella versione SP 1 di XP, utilizzata anche nell'esperimento).

Tra le più importanti sul fronte della sicurezza indubbiamente vi è *Microsoft Windows Firewall*²⁴, in grado di filtrare, bloccare e controllare le connessioni fatte dal sistema operativo o da software installati su esso.

La configurazione di *default* per questo strumento, presente nel SP 2, impedisce la maggior parte delle comunicazioni, al fine di rendere più sicuro il sistema. Ad esempio, uno dei pochi protocolli abilitati è quello *HTTP* in uscita, sulla porta 80, per la navigazione web. Sarà l'utente, successivamente, a dover configurare il firewall *ad hoc* per consentire le comunicazioni da lui ritenute più opportune. Come già specificato, le macchine usata nella prova, risultavano impostate con le regole di default.

Nel caso, però, di Windows Xp SP 1, Linux, MacOS X o lo stesso Windows Server 2003, queste impostazioni permettevano di utilizzare la connessione ad Internet per la maggior parte dei servizi resi disponibili dai protocolli esistenti²⁵; come visto, questa non è la stessa situazione di un Windows XP SP 2 di default, che consente di comunicare solo con pochissimi protocolli (tra i quali, navigazione web e posta elettronica), a meno che l'utente decida diversamente. Questa configurazione, assolutamente efficiente per la protezione della macchina, rende meno affidabili i dati risultanti

24 Con il termine "*firewall*" si indica un strumento, software o hardware che sia, utilizzato per operare sul traffico di dati in una rete. Tramite un'adeguata configurazione, è possibile bloccare determinati tipi di comunicazione (ad esempio, per impedire in ambito aziendale l'utilizzo di software per la *chat*). E' uno degli strumenti più semplici e efficaci per un primo intervento di protezione di una rete o di una singola macchina.

25 La maggior parte delle distribuzioni Linux, ad esempio, hanno attivato di default il server web (da non confondersi con la possibilità di navigare siti: il server web è atto ad *ospitare siti* consultabili tramite rete), SSH, FTP, NNTP, SMTP, POP, VNC. Come spesso sostenuto nei testi riguardanti la sicurezza informatica, quanto più una macchina è ricca di software o servizi, maggiori saranno le probabilità di sfruttare una falla o un bug in essi e, quindi, maggiori saranno le vulnerabilità cui sarà soggetta.

dall'esperimento di Mitnick, relativi a Windows XP SP 2. A nostro parere, quindi, questo aspetto deve essere considerato al fine di valutare al meglio i risultati dell'esperimento in esame.

Il secondo studio è stato condotto da *Honeynet Project*, un gruppo di ricerca sulla sicurezza informatica, il cui obiettivo è quello di diffondere contenuti utili ad amministratori di sistema, o semplici appassionati, per proteggere i propri sistemi.

L'analisi, intitolata “*Know Your Enemy: Why is no one hacking Linux anymore?*”, è stata condotta con una tecnica simile a quella svolta da Mitnick nell'esperimento precedente, ma con il focus principale sulla verifica della sicurezza dei sistemi Linux. I computer utilizzati, infatti, localizzati in otto paesi diversi (USA, India, Inghilterra, Pakistan, Grecia, Portogallo, Brasile e Germania), erano principalmente sistemi Linux, installati con le configurazioni di default e con, in aggiunta, una serie di servizi²⁶ attivati successivamente su quei sistemi in cui questi non erano abilitati dallo sviluppatore, al fine di rendere il test omogeneo tra le diverse macchine. Così come nel caso della ricerca di Avantgarde, le macchine sono state configurate con password di accesso banali.

²⁶ Più precisamente: *FTP*, *SSH*, *HTTPS*, *SMB*. Motivazione di questa scelta, che inevitabilmente rende un sistema più vulnerabile, era quello di avvicinarsi, quanto più possibile, alla configurazione più comune per macchine server collegate ad Internet. Di seguito, una rapida descrizione dei servizi in questione. *FTP*: protocollo per la trasmissione di file: le comunicazioni non sono cifrate. *SSH*: protocollo per il controllo remoto di una macchina in cui le trasmissioni sono completamente cifrate; di norma utilizzato da amministratori di sistema per controllare macchine remote. *HTTPS*: variante di *HTTP*, il protocollo per la navigazione web. In questa implementazione vengono cifrate tutte le trasmissioni e di norma sono utilizzati per la comunicazione di dati sensibili, ad esempio dati di una carta di credito, sul web. *SMB*: software open source per la condivisione di file e stampanti tra macchine Windows e Linux, nato come implementazione per Linux del prodotto analogo Microsoft.

L'esperimento, terminato nel dicembre 2004, è durato sei mesi, durante i quali sono state monitorate tutte le macchine per verificare gli attacchi subiti e le eventuali compromissioni. Da questo studio è emerso un drammatico aumento, rispetto agli anni precedenti, degli attacchi automatizzati, compiuti da worm e software analoghi. Come confermato anche nel lavoro di Mitnick per Avantgarde, i sistemi informatici collegati ad Internet sono esposti a rischi notevoli, derivanti, soprattutto, dai “tentativi a tappeto” condotti da software appositi. Honeynet, che aveva condotto una ricerca analoga nel 2001/2002, ha potuto constatare proprio la diffusione conosciuta da questi software negli ultimi anni e la frequenza con la quale riescono a compromettere un sistema. Il parco macchine utilizzato comprendeva, oltre ai diversi sistemi Linux di prova, anche delle macchine Windows. Gli autori del lavoro, però, hanno specificato di non poter considerare attendibili i dati derivanti dall'utilizzo di queste ultime perché presenti in minor quantità rispetto a quelle con il sistema operativo libero e non adeguatamente seguite, dato che il focus della ricerca non era concentrato sui sistemi Microsoft.

Scopo del lavoro di Honeynet Project non era di confrontare la robustezza dei due diversi sistemi operativi, bensì quella di verificare l'evoluzione conosciuta da Linux negli ultimi due anni; questa precisazione è stata più volte ripetuta dagli autori stessi dello studio successivamente alla diffusione dei risultati su Internet. L'analisi condotta nel 2001/2002 aveva mostrato

una durata media di un sistema Linux, non *patchato* e senza l'utilizzo di particolari accorgimenti tecnici²⁷ - così come le macchine di prova dell'ultimo esperimento - pari a circa 72 ore, prima di essere compromesso. La ricerca condotta nel dicembre 2004 ha mostrato risultati di 3 mesi di resistenza agli attacchi per i sistemi che alla fine dell'esperimento sono risultati compromessi e 4,1 mesi per i sistemi più robusti e mai compromessi. Un sistema solo ha resistito per più di nove mesi.

Indicativo il confronto con lo studio condotto da Mitnick per Avantgarde.

Le conclusioni cui sono giunti i ricercatori sono sostanzialmente tre.

La prima è una semplice constatazione di come Linux sia diventato nel tempo più sicuro: l'incessante sviluppo del kernel e delle applicazioni più importanti hanno reso il sistema più robusto e meno vulnerabile agli attacchi. Allo stesso modo le configurazioni di default dei servizi più diffusi, e potenzialmente insicuri, sono state modificate al fine di tener conto dei rischi, come visto esponenzialmente aumentati negli ultimi anni, dovuti alla connettività.

La seconda conclusione è inerente ai tipi di attacchi che le macchine dell'esperimento hanno subito: così come nello studio di Mitnick per Avantgarde, i computer non sono stati usati da persone, quindi gli attacchi andati a buon fine non potevano fare leva sull'ingegneria sociale e

²⁷ In primis ci si riferisce alle *patch* di sicurezza e agli accorgimenti utilizzati in *SE Linux*, *Security Enhanced Linux*, versione modificata del kernel Linux e di altri software, o in altre distribuzioni che hanno nella sicurezza il focus principale. *SE Linux* è stato scritto e sviluppato partendo dal kernel standard di Linux; è attualmente utilizzato in importanti infrastrutture informatiche come la NASA e la NSA, National Security Agency degli Stati Uniti, cui si deve la maggior parte dello sviluppo del sistema stesso.

sull'errore o ingenuità umana. L'unico tentativo di questo tipo è stato, anche se solo passivamente, quello del “*password cracking*”, indovinando le semplici parole d'accesso scelte per l'esperimento. Questo punto dovrebbe far riflettere, ancora una volta se necessario, sull'importanza del tema della sicurezza informatica che non costituisce più un rischio per le sole grandi aziende ed infrastrutture, ma un rischio cui tutti i sistemi connessi ad Internet sono esposti.

Infine, i ricercatori di HoneyNet Project sono giunti alla terza, e più ovvia, conclusione secondo la quale la maggior parte degli autori di malware preferiscano scrivere software per il più diffuso bacino d'utenza dei sistemi Windows piuttosto che per il ristretto insieme rappresentato dalla piattaforma Linux. Gli autori concludono affermando che, in vista di una maggiore e sempre più rapida diffusione dei sistemi Linux, sarà molto probabile conoscere un parallelo aumento di software malware per il sistema operativo libero.

2.1.3 Rigorosa gestione della multiutenza

La multiutenza è stata, nello sviluppo di Linux, da sempre una esigenza fondamentale. Nato per essere eseguito su macchine server, campo nel quale è fondamentale tutelare sicurezza e stabilità del sistema, ha seguito la linea tracciata dai sistemi Unix nella gestione, e separazione, dei permessi

accordabili ad ogni singolo utente.

La prima sensibile differenza nell'utilizzo di un sistema Linux rispetto a Windows è costituita dall'impossibilità di eseguire, con un account utente normale, certi tipi di operazioni, comuni sul prodotto Microsoft. Installare applicazioni, modificare file di configurazione, collegarsi ad Internet, o semplicemente spegnere il computer, sono operazioni non consentite all'utente comune²⁸. Si rende così necessaria l'assunzione temporanea dell'identità dell'utente amministratore (l'utente *root* nei sistemi Linux e *Administrator* nei sistemi Windows) al fine di poter eseguire il comando particolare. Un'attenta e lungimirante configurazione del sistema da parte dell'amministratore, insieme all'utilizzo di software apposito²⁹, può consentire, in un ambiente comunque caratterizzato da elevati standard di sicurezza, all'utente non privilegiato l'esecuzione dei comandi più frequenti. Un'altra caratteristica dei sistemi Linux è quella di gestire nativamente i *filesystem*³⁰ *ext2/3*. Questi permettono l'assegnazione, per ogni file, di due “informazioni di possesso” e tre gruppi di permessi. Le informazioni di

28 Una precisazione è d'obbligo: le operazioni in questione non sono permesse all'utente normale quando costituiscono una potenziale minaccia (lo spegnimento di un server, ad esempio, ne interromperebbe il servizio) oppure quando vanno a modificare aspetti *globali* del sistema. Personalizzare il proprio “*desktop*”, ad esempio, è concesso, in quanto relativo solo all'ambiente operativo dell'utente specifico. Allo stesso modo, installare applicazioni nella propria directory personale.

29 Il programma *sudo* (contrazione dell'espressione “*superuser do*”) ne è un esempio. Permette all'amministratore di configurare un sistema in modo tale da permettere un particolare comando (lo spegnimento del computer, ad esempio) ad un utente o ad un gruppo di utenti, in completa sicurezza.

30 Tecnica per la memorizzazione, la lettura e l'organizzazione dei file in un dispositivo. Per poter usare un dispositivo di memorizzazione, infatti, come un hard disk ad esempio, è necessario predisporlo con un particolare filesystem (procedura spesso nota come “formattazione”). Il filesystem standard per Linux è *ext2* e più recentemente la nuova versione *ext3*. Windows, invece, adotta da qualche anno *NTFilesystem*, dopo aver sostanzialmente abbandonato il precedente e obsoleto *FatFilesystem*.

possesso sono utilizzate per assegnare la proprietà di un file ad uno specifico utente e ad un gruppo al quale egli appartiene; i gruppi di permessi, invece, sono tre, ognuno caratterizzato dai permessi in lettura, scrittura ed esecuzione e sono attribuiti in relazione a: proprietario, gruppo e altri utenti.

Un esempio potrebbe meglio illustrare i vantaggi di questa funzionalità. Un file potrebbe appartenere all'utente "Rossi" del gruppo "Contabilità" (informazioni di possesso); l'utente proprietario, Rossi, potrebbe avere libero accesso in lettura, scrittura ed esecuzione su quel file. Una gestione dei permessi al fine di permettere la collaborazione tra colleghi, però, potrebbe consentire a tutti gli utenti appartenenti al gruppo "Contabilità" (i colleghi di reparto di Rossi) di leggere e modificare il contenuto del file, mentre i colleghi non appartenenti a quel reparto (gli impiegati nel reparto marketing, ad esempio) potrebbero avere accesso solo in lettura, ma non in modifica.

Una delle prime conseguenze legate ad una severa, e soprattutto rigorosa, gestione dei permessi è quella di limitare i danni in caso di infezioni di malware o in caso di errore umano, entrambi da parte di un utente non privilegiato. Se un utente di questo tipo esegue un programma, quest'ultimo avrà i diritti ed i permessi dell'utente che lo ha avviato; effetto diretto di questo comportamento è che se un worm, attivato ad esempio tramite incauta lettura di un allegato di posta, provasse a rimuovere dei file o ad

infettarne altri (quindi, modificarli), riceverebbe un errore di “Permesso negato”, a meno che non siano dati accessibili all'utente “di partenza”. In questo modo, un worm attivato da un utente non privilegiato, potrà arrecare danni solo ai file cui ha accesso, presumibilmente pochi e non fondamentali per il sistema. Allo stesso modo non potrà interrompere un servizio, non potrà accedere a determinate risorse, non potrà modificare file di configurazione del sistema.

Un comportamento del genere da parte del sistema operativo porta inevitabilmente ad un “ambiente a comportamenti stagni”: l'utente non privilegiato semplicemente *non può* arrecare danni all'intero sistema, causando, ad esempio, un blocco del sistema; in caso di infezione o di errore umano, sarà sufficiente rimuovere l'account dell'utente responsabile e ricrearlo ovvero provvedere direttamente alla rimozione dei software responsabili. Al fine di creare un sistema che sia *effettivamente* a compartimenti stagni, però, sarà necessaria un'attenta configurazione dei permessi, degli utenti e dei gruppi da parte dell'amministratore.

2.1.4 Distribuzione di *patch* e *bugfixing*

Come più volte accennato, l'esistenza di bug e problemi di sicurezza in un software è una evenienza quasi inevitabile; l'errore umano nella programmazione è sempre possibile. Analogamente, risulta molto difficile

per un singolo sviluppatore, o per un piccolo team, procedere ad un accurato *testing* del software per identificare problemi o falle nel codice. L'attività svolta da un numero maggiore di individui è, inevitabilmente, più accurata e ad ampio raggio.

Proprio per queste motivazioni, diventa sempre più importante procedere, una volta scoperto e risolto un bug, alla tempestiva distribuzione delle patch apposite per risolvere il problema. Ritardi nella creazione di questi software “risolutori” possono portare, ed è quanto successo in molti casi, a gravi problemi di sicurezza per tutti quei sistemi esposti, in particolar modo per tutti quei computer collegati alla Rete e non ancora dotati degli aggiornamenti. Esistono portali web appositi³¹ per la diffusione di informazioni riguardo bug e, soprattutto, falle di sicurezza nei software più diffusi: risorse di questo tipo vengono di solito utilizzate da amministratori di sistema per venire a conoscenza di problemi di sicurezza sorti negli applicativi da loro utilizzati ed, eventualmente, provvedere. Naturalmente la libera accessibilità a queste informazione rende queste risorse disponibili anche a chi volesse sfruttare, al fine di colpire determinati sistemi, le falle di sicurezza più “interessanti” e non ancora risolte. La tempestiva pubblicazione delle specifiche dei bug e la ritardata distribuzione delle patch di aggiornamento diventano, quindi, un pericoloso mix per la sicurezza di qualsiasi sistema esposto a rischi esterni e, in primo luogo, per tutte le macchine collegate ad Internet.

³¹ Cfr. www.securityfocus.org e www.bugtraq.org per gli esempi più importanti.

Linux su questo piano gode, a nostro avviso, di un netto vantaggio rispetto ad un qualsiasi produttore di software proprietario: la comunità di sviluppatori che lavora sul progetto è notevole ed è così in grado di garantire un adeguato testing. Appositi team di sviluppo del kernel di Linux lavorano *esclusivamente* alla risoluzione dei bug, alla creazione delle patch ed alla loro distribuzione. L'attenzione di questa comunità da sempre rivolta alla sicurezza e stabilità dei sistemi informatici, quindi, porta il software open source ad un vantaggio rispetto ai concorrenti difficilmente colmabile dalle aziende presenti nel settore.

Nel paragrafo 2.1 emerge che in alcuni casi di infezione gli stessi server di Microsoft e le sue infrastrutture, si sono rivelate vulnerabili agli attacchi e colpite. In questo caso ci riferiamo agli worm che sfruttavano bug e falle di sicurezza dei software, note e potenzialmente risolvibili con patch. Nonostante fossero state già distribuiti gli aggiornamenti dalla Microsoft stessa per risolvere i problemi di sicurezza, molti dei sistemi dell'azienda di Redmond, ivi compresi server web e database server, non erano stati aggiornati e si presentavano quindi vulnerabili. Furono colpiti dall'infezione e resi inutilizzabili. Semplice distrazione ed inefficienza o difficoltà di aggiornamento?

Nell'agosto del 2004, lo staff IT della IBM diffonde un comunicato³² con il quale invita i dipendenti a non procedere all'aggiornamento di Windows XP

³² Fonte: Comunicato “*To patch – or not to patch*”, 6 agosto 2004, pubblicato su Intranet locale IBM.

con il tanto atteso Service Pack³³ 2 (Microsoft XP SP2), uno dei “prodotti” più attesi dell'anno. Microsoft in quelle settimane, infatti, stava pubblicizzando la distribuzione di SP2 e, dopo le recenti e gravi infezioni di worm registrate nel corso dell'anno, spingeva i propri utenti a procedere rapidamente con l'aggiornamento al fine di rendere i sistemi Windows XP più sicuri. L'utente aveva la possibilità di effettuare il *download* del file da Internet ovvero richiedere la consegna per posta, in tutto il mondo, gratuitamente.

Il Service Pack 2, però, non poteva essere assolutamente considerato semplicemente un “insieme di patch” (*roll-up patch*): modificando profondamente il sistema, rimuovendo funzionalità ed installandone di nuove, si configurava in definitiva come un vero e proprio aggiornamento dell'intero sistema operativo. Così come risultò successivamente, Microsoft fu costretta a compiere un *trade-off* tra sicurezza e compatibilità: la scelta dell'azienda di Redmond si è orientata, probabilmente non a torto, sulla prima soluzione, creando però tutta una serie di problemi di compatibilità tra applicazioni esistenti per Windows e il nuovo Service Pack. Gli stessi prodotti Microsoft, come ad esempio Microsoft CRM 1.2 (*Customer Relationship Management*), presentavano incompatibilità che potevano essere risolte solo dopo un ulteriore aggiornamento di questi software, con patch non ancora distribuite.

³³ Insieme di aggiornamenti, distribuiti in unico file contemporaneamente. Utilizzati di solito da Microsoft per evitare che l'utente si colleghi ad Internet per reperire decine di patch diverse, raccolgono in unica distribuzione diversi aggiornamenti per risolvere bug, falle di sicurezza e per implementare nuove funzionalità.

Il caso IBM, uno dei più importanti clienti dell'azienda di Redmond con più di quattrocentomila computer desktop installati con Windows XP, verteva sulle stesse difficoltà: nel comunicato si legge, infatti, che le applicazioni “*high profile*” e “*business-critical*” utilizzate da IBM non sono compatibili con Service Pack 2 e, quindi, non si può procedere all'aggiornamento, se non bloccando l'intera attività legata a queste applicazioni.

Il documento proseguiva affermando che l'azienda si sarebbe attivata personalmente al fine di modificare l'aggiornamento, fornendone una versione *ad hoc* compatibile; solo a partire da quel momento sarebbe stato opportuno, e sicuro, procedere all'installazione del Service Pack 2. Anche IBM fu costretta ad eseguire un *trade-off* tra sicurezza e compatibilità; fu così obbligata, a scegliere la seconda soluzione, onde evitare una completa interruzione delle attività che facevano uso delle applicazioni “business-critical”. I sistemi dell'azienda rimasero esposti ad attacchi fino alla distribuzione del Service Pack 2 personalizzato, creato dalla stessa IBM, e distribuito qualche settimana dopo.

Questi episodi mostrano come la tempestività nella distribuzione delle patch non sia sufficiente a garantire una soluzione a questo tipo di problemi: strettamente necessaria si rivela anche la *qualità* delle patch. La valutazione qualitativa sulla “bontà” del software di aggiornamento di due diversi sistemi operativi risulta molto difficile, ma un dato interessante è quello che rappresenta i problemi legati all'utilizzo di un determinato

aggiornamento (come quelli precedentemente riportati per Service Pack 2) e la numerosità dei bug e delle falle scoperte, tutti possibili “obiettivi” di nuove patch.

Riguardo quest'ultimo punto, abbiamo analizzato uno studio di Coverity³⁴, azienda indipendente specializzata nel controllo del codice di applicazioni, in merito alla ricchezza di bug (“*bug density*”) nel kernel di Linux.

Dopo uno studio durato quattro anni sui 5,7 milioni di righe di codice che costituiscono il “cuore” del sistema operativo libero, gli analisti hanno trovato 985 bug. Un software commerciale analogo, in quanto a complessità (sostanzialmente rappresentato dal numero di linee di codice), presenta circa 5000 falle. Quest'ultimo dato è confermato da uno studio della *National Cybersecurity Partnership's Working Group on the Software Lifecycle*, attiva nel campo dello sviluppo di applicazioni per promuovere una maggior attenzione sulla sicurezza del software. L'analisi porta ad una stima di 1-7 bug per ogni mille righe di codice di un generico software proprietario; utilizzando queste proporzioni, il kernel di Linux dovrebbe contenere da 5700 a 40000 falle, contro le circa 1000 rilevate.

Non è stato possibile applicare lo studio analogo sulla “bug density” del codice di Windows data l'impossibilità di disporre del codice sorgente del sistema operativo Microsoft.

³⁴ Coverity, “*Analysis of the Linux kernel*”, disponibile su www.coverity.com

2.1.5 Sicurezza nel browser

Uno dei “successi” più recenti del software open source ha avuto luogo nel campo dei browser web. Assoluto protagonista di questo particolare e giovane mercato è Microsoft Internet Explorer. Software utilizzato dalla quasi totalità degli utenti Internet, è stato protagonista negli anni Novanta di uno dei più grandi successi di Microsoft; l'azienda di Redmond, infatti, entrò nel mercato solo in un secondo momento, quando protagonista era la *Netscape Communications Corporation* con il suo browser Netscape Navigator, detentore in quegli anni della maggior parte del mercato.

A partire dal 1996, le due aziende si fronteggiarono in quella che divenne “la guerra dei browser” e che fu caratterizzata da un incessante sviluppo di entrambi i prodotti. I due software si arricchivano di funzionalità sempre più importanti nella navigazione dei contenuti web e progressivamente la quota di mercato di Internet Explorer aumentava. Uno dei punti di forza del browser Microsoft, oltre alle funzionalità da esso offerte, era l'incessante sviluppo che l'azienda di Redmond poteva garantire; il concorrente Netscape, una piccola azienda a confronto del colosso mondiale, non era assolutamente in grado di garantire un così incessante ritmo.

Un altro strumento con in quale Microsoft riuscì ad imporre il proprio prodotto fu quello di offrirlo a corredo del sistema operativo Windows e ad integrarlo con esso. Questa strategia, che come visto interessò anche le corti

dei tribunali³⁵ e portò l'azienda di Redmond ad un importante risarcimento, riuscì ad imporre Internet Explorer in un intervallo di tempo molto breve. Essendo installato su tutti sistemi Windows, e proposto come browser predefinito, per la maggior parte degli utenti costituiva anche *l'unico* strumento possibile per navigare il web ed indubbiamente questo contribuì a far perdere terreno a Netscape Navigator. La situazione che vedeva Internet Explorer utilizzato dalla quasi totalità degli utenti Windows è rimasta tale fino alla metà del 2004 circa. In questo periodo due importanti fenomeni si stanno diffondendo: da un lato i problemi di bug, sicurezza e carenza di funzionalità del prodotto Microsoft e dall'altro lato la qualità raggiunta da un progetto concorrente, *Mozilla Firefox*.

Nato dalla Mozilla Foundation, creata al fine di sviluppare il codice sorgente rilasciato dalla Netscape Corporation dopo essere stata sconfitta da Microsoft, questo browser web si era già imposto come ottimo strumento nella comunità Linux, ma stentava a diffondersi tra gli utenti Windows. Grazie a funzionalità non presenti in Internet Explorer, ai problemi di sicurezza di quest'ultimo³⁶ e alla qualità generale del progetto, Mozilla Firefox è riuscito nel volgere di pochissimo tempo ad ottenere visibilità e ad aumentare la sua presenza sul mercato³⁷. Il browser della Mozilla

35 Cfr. § 1.1.

36 Questi tipi di problemi hanno portato importanti istituzioni a sconsigliare l'utilizzo del browser Microsoft per ragioni di sicurezza. Il CERT statunitense (*US-Computer Emergency Readiness Team*), il Ministero della Comunicazione finlandese costituiscono alcuni esempi: con alcuni comunicati delle rispettive istituzioni si consigliava di non utilizzare Internet Explorer al fine di evitare il diffondersi di infezioni di alcuni worm come "*Bofra*". Fonte: <http://www.kb.cert.org/vuls/id/713878>

37 Dall'inizio di dicembre 2004 (il lancio ufficiale della versione per Windows è stata nel novembre dello stesso anno) alla metà di gennaio 2005 Firefox è arrivato al 4,78% del mercato.

Foundation costituisce uno dei successi più importanti degli applicativi open source e costituisce, tra l'altro, uno degli strumenti per effettuare una lenta migrazione verso sistemi Linux.

2.1.6 Modularità del kernel

Il sistema operativo Linux, come già anticipato, è nato come frutto di una collaborazione del Progetto GNU della Free Software Foundation e di Linus Torvalds, creatore del kernel. Quest'ultimo, elemento principale del sistema operativo, fu disponibile solo diversi anni dopo il termine del lavoro condotto dal team di sviluppo GNU, costituendo così il tassello mancante del progetto. Con il codice scritto da Linus Torvalds si era finalmente di fronte ad un sistema operativo completo. E libero.

Da qui nacque la diffusione (anche al di fuori dell'originaria comunità *hacker*) ed il successo dell'open source come modello di sviluppo del software e il panorama di Linux si arricchì di nuovi e importanti progetti.

La scelta di Torvalds fu quella di progettare un'architettura modulare³⁸ del kernel, rendendo possibile la configurazione (e la compilazione) in grado di adattarsi alle più svariate architetture (qui inteso con riferimento alle caratteristiche hardware) ed esigenze. Risultato di questa architettura di

Internet Explorer ha nello stesso periodo perso lo 0,7%, attestandosi, primo assoluto, su 92,7% degli utenti Internet. Nel giugno 2004 il browser Microsoft era accreditato per detenere il 96,7% del mercato; nell'arco di sette mesi, quindi, si è registrata una diminuzione del 4%.
Fonte: www.websidestory.com

38 Costituito da un insieme di moduli necessari all'implementazione di determinate funzioni. Ad esempio, un modulo che funziona come “driver” per una scheda video o per un dispositivo di memorizzazione.

sistema è che Linux si presta ad essere usato su sistemi dalle caratteristiche molto diverse: da sistemi desktop dalla potenza ridotta, a server con maggiori capacità, dai mainframe per elaborare grandi quantità di calcoli ai computer portatili. Uno dei vantaggi di usare in un network aziendale un sistema operativo così modulare è quello di poter configurare le singole macchine secondo le esigenze più opportune, arrivando ad un livello tale di personalizzazione del sistema operativo non raggiungibile con Windows.

La configurazione in questione non si limiterebbe alla scelta dei software da installare o dei permessi accordati al singolo utente; modificando opportunamente la configurazione del kernel, si è in grado di rendere *impossibile* l'esecuzione di determinate operazioni, agendo al livello base di supporto del sistema operativo. Successivamente affronteremo il tema della modularità del kernel Linux relativamente alla sicurezza informatica e alle sue implicazioni.

Un esempio può illustrare meglio questo concetto. Si prenda il caso delle postazioni non appartenenti ad uno specifico utente, ma utilizzate da diversi soggetti. E' possibile fare in modo che una macchina non possa *tecnicamente* accedere ai dispositivi di memorizzazione di un network aziendale: agendo sulla configurazione del kernel, è possibile aggirare il problema della limitazione a livello software dei permessi attribuiti ad una postazione. Una restrizione a livello software, infatti, è intrinsecamente soggetta a vulnerabilità legate alla natura stessa dell'operazione; un attacco

andato a buon fine, che permetta all'aggressore di assumere i *diritti di root*, ovvero il pieno potere sulla macchina, potrebbe esplorare le risorse condivise della rete e tentare l'intrusione verso altre macchine.

Agendo a livello di configurazione del kernel, invece, si ottiene una postazione che *non può*, neanche con i privilegi di amministratore, accedere alle risorse condivise del network. La tecnica per ottenere un simile risultato è molto semplice e facilmente replicabile per essere applicata su una moltitudine di postazioni. E' sufficiente agire sulla ricompilazione del kernel, una delle caratteristiche più importanti in tema di modularità di Linux.

Questa operazione consiste nel ricostruire integralmente, partendo dal codice sorgente, il cuore stesso del sistema operativo; configurando ad hoc il sorgente del kernel è possibile “abilitare” o “disabilitare” determinate funzionalità del sistema che si andrà a costruire. Durante la ricompilazione del kernel, infatti, l'amministratore potrà decidere quale delle funzionalità includere nel kernel (come, ad esempio, il supporto USB, il supporto ai filesystem di rete per l'accesso alle directory condivise, supporto alla scheda video).

In questo modo, ne risulta un sistema operativo che non è *in grado* di eseguire certe operazioni, esulando quindi dalla configurazione dei permessi. In particolari configurazioni di sistema, questo può essere un potente strumento al fine di proteggere una postazione e, parallelamente,

tutta la rete ad essa collegata.

2.2 Flessibilità

La flessibilità, nella “configurazione” del sistema operativo e del suo kernel come appena visto, è una caratteristica anche della maggior parte del software per Linux.

I file di configurazione dei diversi applicativi sono normalmente dei semplici file di testo, al cui interno sono presenti tutte le opzioni, in forma di stringa, supportate dal software. All'utente è sufficiente leggere il file di configurazione e personalizzare così il funzionamento e l'aspetto dell'applicativo; questa filosofia di progettazione, derivata dai sistemi Unix, è stata la più diffusa per anni e permette, semplicemente copiando il file di configurazione, un'importazione veloce delle proprie impostazioni personali su più macchine. Questo aspetto è sicuramente un vantaggio nel caso di sistemi costituiti da un gran numero di personal computer che devono presentare le stesse impostazioni del software installato, come, ad esempio, nel caso di un network aziendale.

Rispetto ai software di tipo *closed source*, la personalizzazione offerta dal software libero è di norma superiore, garantendo accesso alla totalità delle possibilità offerte dal software; anche in questo caso, il comportamento del programmatore open source in sede di sviluppo è dettato principalmente

dalla volontà di consentire la collaborazione su uno stesso progetto di più individui. Il metodo più efficiente diventa, quindi, quello di consentire all'utente di personalizzare quanto più possibile il comportamento del programma o poterne meglio capire il funzionamento, spesso grazie anche alla “documentazione” inserita nello stesso codice sorgente. Un approccio di questo tipo da parte degli sviluppatori è anche motivato dal target che Linux ha avuto, ed in parte ancora ha: i potenziali utenti sono soggetti che difficilmente troverebbero sufficiente una configurazione superficiale e poco approfondita e riterrebbero fondamentale avere il controllo pressoché completo dell'applicazione. A distanza di anni, la filosofia *hacker* continua ad avere una notevole influenza sullo sviluppo di software per Linux.

La flessibilità illustrata fino a questo momento è principalmente riferita alle applicazioni o al kernel, nei limiti concessi dai rispettivi sviluppatori; se gli autori di un software, scrivendone il codice, non rendono possibile una determinata operazione o comportamento, l'utente non potrà fare niente per aggirare questo ostacolo. In realtà una situazione di questo tipo è riferibile esclusivamente al software closed source: nel mondo open source, se un soggetto ne ha le capacità, può intervenire a piacimento sul codice sorgente di un'applicazione e modificarne comportamento e funzionalità. Questo può costituire un notevole vantaggio per le realtà che fanno uso di questo tipo di software; fino a qualche anno fa, la scelta nell'adozione di un software era limitata a quanto disponibile sul mercato o, qualora le alternative non

fossero state all'altezza, a sviluppare in proprio un'applicazione, nel caso rivolgendosi a professionisti esterni. Naturalmente la creazione e lo sviluppo *ex novo* di un'applicazione, soprattutto se complessa, necessita di un notevole impiego di risorse; nel software open source la situazione è completamente differente. Se un'azienda ritiene un software valido, anche se non perfettamente conforme alle proprie esigenze, può lavorare sul codice sorgente, modificandone funzionalità e comportamento secondo le proprie necessità. Una potenzialità del genere può, da sola, rendere conveniente la migrazione a Linux del network o di parte di esso.

2.3 Facilità nella manutenzione dei sistemi

Le attuali caratteristiche di Linux, come già visto in tema di sicurezza e flessibilità, sono frutto del particolare target cui per anni è stato destinato; anche nel caso della manutenzione dei sistemi il sistema operativo libero ha caratteristiche che possono agevolare molto il lavoro di amministratori responsabili per grandi network, risparmiando tempo e risorse e permettendo un servizio di miglior qualità e dalla maggior continuità.

Per "manutenzione dei sistemi" qui si intende il periodico controllo delle macchine da monitorare, l'aggiornamento dei software installati sulle macchine e determinate operazioni, come ad esempio il *backup* dei dati, necessarie in particolari configurazioni aziendali.

Le caratteristiche che rendono Linux estremamente valido in questo particolare momento dell'amministrazione di sistemi informatici sono, a nostro avviso, principalmente due:

- accesso da remoto e controllo totale sulla macchina
- shell scripting

Per il primo si intende la possibilità per un utente di collegarsi da remoto ad una macchina, utilizzando software e protocolli appositi (*SSH* ne è l'esempio più diffuso) e svolgendo operazioni come se fosse davanti al personal computer. Importantissimo nel caso di grandi network, diventa assolutamente fondamentale nel caso di gestione di macchine *headless*, ovvero senza monitor, come nel caso di macchine server. Utilizzando SSH per connettersi ad una macchina remota, i comandi vengono impartiti tramite interfaccia da linea di comando³⁹ ed è possibile eseguire tutte le operazioni normalmente svolte dalla postazione reale.

Questa funzionalità non è presente nei sistemi operativi Windows che *necessitano* dell'interfaccia grafica per eseguire la quasi totalità delle operazioni; questa lacuna è facilmente spiegabile con il processo di

³⁹ In questo caso l'interazione con il computer avviene tramite inserimento di comandi, seguendo una determinata sintassi propria della *CLI*, *Command Line Interface*. Con la diffusione dei personal computer, l'utente si è abituato alla più intuitiva *GUI*, *Graphical User Interface*, con la quale l'interazione avviene tramite mouse, tastiera e interfaccia grafica caratterizzata da icone e finestre. In Linux il controllo della macchina può essere eseguito in entrambi i modi, fatta eccezione naturalmente per tutti quei software che utilizzano esclusivamente l'interfaccia grafica. Nel caso di gestione di server, l'interazione tramite interfaccia grafica si rivela lenta, superflua e non efficace.

sviluppo che il prodotto in questione ha conosciuto e per il target di utenza del software Microsoft. Ciò nonostante esistono degli strumenti di amministrazione remota anche per Windows⁴⁰ che permettono una sessione remota di tipo esclusivamente grafico con la quale, anche in questo caso, è possibile eseguire qualsiasi tipo di operazione sul computer. Un controllo remoto di questo tipo, però, presuppone una serie di *input* continui da parte dell'amministratore che deve eseguire le operazioni dall'inizio alla fine del processo, attendendone la conclusione. Nel caso di amministrazione di più macchine, e qualora si renda necessario lo svolgimento di una medesima operazione su più sistemi, come ad esempio nel caso di aggiornamento del software e backup dei dati, sarà necessario svolgere la stessa operazione più volte.

In queste situazioni l'utilizzo dello "*shell scripting*"⁴¹ renderebbe facile e automatizzabile lo svolgimento, periodicamente se necessario, di una serie di comandi, con la possibilità di replicarlo su più macchine. L'intervento dell'amministratore, in questo caso, si limiterebbe alla creazione, *una tantum*, dello script, all'eventuale configurazione di software appositi per la ripetizione dei comandi⁴² ed alla lettura periodica delle email inviate dalle

40 Tra questi assolutamente degno di nota *VNC*, ideato e sviluppato nei laboratori *Olivetti & Oracle Research Lab*, prima di essere rilasciato come software open source e ripreso dal Dipartimento di Ingegneria dell'Università di Cambridge ed, attualmente, da una azienda legata ad AT&T. E' disponibile per tutti i più importanti sistemi operativi, ivi compresi Windows e Linux.

41 Uno *script* è un semplice file di testo nel quale, con una particolare sintassi, vengono inseriti diversi comandi da eseguire, contemporaneamente o uno dopo l'altro. Un classico esempio potrebbe essere quello di: collegarsi ad una macchina remota, effettuare il login, procedere con l'aggiornamento del software, effettuare il backup e notificare tramite email all'amministratore il completamento con successo dei comandi o eventuali disservizi.

42 Ad esempio, *cron*, software utilizzato da sistemi Linux e Unix like per l'esecuzione periodica di operazioni.

diverse macchine al fine di identificare eventuali malfunzionamenti. Anche in assenza di operazioni da svolgere periodicamente, il controllo completo di una macchina da remoto può rendere possibile un servizio di assistenza remota che, altrimenti, risulterebbe inapplicabile per via delle risorse necessarie per la gestione di un network complesso; l'utente che si trovi in difficoltà con una particolare applicazione, o che scopra un malfunzionamento della macchina, può contattare il servizio tecnico che può intervenire *direttamente* sulla postazione e provvedere a prestare assistenza o identificare e risolvere il problema.

L'aggiornamento del software, inoltre, può avvenire tramite strumenti particolari di pacchettizzazione ed installazione. Le diverse distribuzioni Linux, infatti, utilizzano spesso differenti strumenti software per l'installazione, configurazione e distribuzione dei programmi disponibili. La distribuzione di riferimento che utilizzeremo in questo lavoro è *Debian GNU/Linux*, una delle prime distribuzioni create, insieme a *Slackware*⁴³.

Caratterizzata dall'essere costituita *esclusivamente* da software libero con licenza GPL, è considerata una delle migliori distribuzioni per installazioni server. Debian usa anche un sistema di gestione pacchetti tra i più efficienti in circolazione, *dpkg* e *apt-get*.

Questo strumento si collega ad alcuni server su internet, dove vengono memorizzati tutti i programmi e gli aggiornamenti, controlla cosa sia disponibile e aggiorna (o installa) automaticamente il sistema, risolvendo le

⁴³ Cfr: www.debian.org e www.slackware.org

dipendenze necessarie⁴⁴. Una funzionalità di questo tipo rende estremamente rapido l'aggiornamento, ad esempio per motivi di sicurezza, di interi sistemi costituiti da decine di macchine: un semplice script, contenente i comandi per la connessione remota alle macchine, l'aggiornamento dei pacchetti e il backup finale dei dati, può essere scritto in pochi secondi ed essere utilizzato su tutte i computer presenti in un network.

2.4 Riduzione dei costi legati ai client Linux

La riduzione dei costi legati alla gestione e configurazione di macchine desktop con Linux sono, a nostro avviso, riconducibili a più aspetti:

- a) costi di licenza del sistema operativo
- b) costi di licenza del software aggiuntivo
- c) costi per l'acquisizione di hardware
- d) costi per la manutenzione dei client

I primi due punti sono strettamente legati alla natura stessa del software

⁴⁴ A causa della struttura fortemente modulare del sistema operativo Linux, le dipendenze da soddisfare per installare un determinato pacchetto possono costituire un problema serio nella gestione del software da parte dell'utente amministratore, richiedendo notevole dispendio di tempo. In questo caso, l'utilizzo di un sistema di pacchettizzazione come *apt* di Debian, risolvendo automaticamente tutte le dipendenze necessarie, può rendere più veloce ed efficiente la configurazione di un sistema. Lo strumento in questione non è l'unico in grado di compiere questo tipo di operazione: *urpmi* di Mandrake, distribuzione francese di Linux, costituisce un altro esempio.

preso in esame, il software commerciale. Il punto a) si riferisce *esclusivamente* al costo sostenuto da un'azienda per l'acquisizione delle licenze del sistema operativo. Saranno quindi esclusi altri oneri legati all'utilizzo di software aggiuntivo. Quest'ultimo, qui indicato al punto b), costituisce un'altra importante fonte di spesa. Il punto c) si riferisce agli oneri affrontati per l'acquisizione dell'hardware necessario alle macchine desktop. Il punto d) , infine, fa riferimento alle operazioni di gestione e manutenzione dei computer nel corso del tempo ed i costi ad esse associati.

2.4.1 Costi di licenza per il sistema operativo

Il sistema operativo è un insieme di software e librerie che permette ai diversi componenti di un computer di dialogare tra loro e all'utente di interagire con essi.

Nonostante sia uno strumento fondamentale per l'utilizzo di un personal computer, le funzioni che permette di svolgere, in assenza di altri programmi, sono poche e variano dall'esplorazione del filesystem alla gestione dei diversi protocolli o al movimento dei file.

L'allestimento di un network informatico, composto da diversi personal computer, dovrà quindi comprendere una spesa fissa per ogni macchina⁴⁵.

Un ulteriore problema, anche in questo caso legato non ad un particolare

⁴⁵ Attualmente, il costo di una licenza singola di Windows XP varia dai 90€ per la versione "Home" e 140€ per "Professional". Non sono in questo caso considerate promozioni temporanee, né riduzioni per rivenditori, né sconto per l'acquisto in "multilicenza".

prodotto ma alla natura stessa del software commerciale, è quello degli aggiornamenti di versione. Nello specifico non ci riferiamo agli aggiornamenti di sicurezza o a quelli di lieve entità (*minor upgrades*), normalmente resi disponibili gratuitamente tramite Internet dallo stesso produttore, quanto al cambiamento della versione stessa del software, come nel caso del passaggio da Windows 2000 a Windows XP. In casi di questo tipo si rende necessario l'acquisto di nuove licenze, pari al numero di macchine da aggiornare, con un'ulteriore spesa non quantificabile a priori. Un periodo di tempo ragionevole entro il quale considerare necessario l'aggiornamento è a nostro avviso di circa cinque anni.

E' doveroso fare però una precisazione a riguardo: normalmente la scelta di aggiornare il sistema operativo o meno non è una decisione *semplicemente discrezionale*, perché spesso si pongono problemi di supporto da parte dello sviluppatore e di compatibilità. Il primo ordine di problemi si concretizza solitamente con l'impossibilità di aggiornare il sistema operativo per proteggersi da bug, falle di sicurezza o semplicemente per usufruire dei *minor upgrades*. Altro tipo di difficoltà, invece, viene posta dalla compatibilità delle nuove applicazioni o del nuovo hardware con i sistemi operativi più datati (“retrocompatibilità”) e che rende, di fatto, impossibile utilizzare determinati strumenti.

Il mondo dell'informatica conosce una così rapida e *continua* evoluzione che è difficile poter scegliere, senza inconvenienti di sorta, di “rimanere

indietro”. La scelta, quindi, si concretizza di solito con la *necessità* di aggiornare, dopo un certo periodo di tempo, il proprio sistema operativo e le applicazioni e sostenerne, quindi, i relativi costi di licenza⁴⁶.

Nel mondo Linux e del software libero in generale non è *tutto* gratuito. Ricordando che questo tipo di software è “*free as in speech, not as in beer*”, è necessario superare appunto il luogo comune che sia tutto disponibile gratuitamente. Il kernel Linux, il cuore del sistema operativo, è liberamente accessibile su Internet⁴⁷ così come la maggior parte del software ad esso dedicato, ma non è una condizione *sine qua non* per poter essere considerato software libero, come già affrontato in precedenza⁴⁸.

Un fenomeno molto diffuso, inoltre, è stato la nascita di diversi gruppi di programmatori, o addirittura società, che hanno deciso di sviluppare, e rilasciare, la propria “distribuzione Linux”. In questo caso il lavoro svolto nella “personalizzazione” di Linux, dal punto di vista tecnico, consta principalmente in strumenti software per la configurazione del sistema, per l’iniziale installazione, o per altri aspetti minori, ma non ha in nessun modo a che vedere con il kernel⁴⁹. Nel campo delle distribuzioni che prevedono costi di licenza gli esempi più importanti sono *Red Hat Software*, azienda statunitense, e in Europa la tedesca *SUSE*, recentemente acquisita da

46 Ci soffermeremo meglio su questo aspetto nel § 2.5, dedicato al *vendor tie-in*.

47 Il sito di riferimento è www.kernel.org

48 Cfr. § 1.2.1 e § 1.2.2.

49 Il kernel utilizzato nelle diverse distribuzioni è in linea generale il medesimo. Talvolta, però, le singole distribuzioni decidono di applicare delle patch, ad esempio, per il montaggio automatico dei dispositivi o per altre funzionalità minori. Il cuore del sistema operativo, ad ogni modo, rimane sempre il medesimo.

Novell.

Le due distribuzioni in esame, pur essendo vendute nei negozi o tramite Internet, sono liberamente disponibili sul sito delle rispettive aziende, lasciando al singolo utente la decisione se effettuare il download ovvero comprare l'intero pacchetto. Ciò che Red Hat e SUSE vendevano, e vendono, *non è il sistema operativo*, ma un insieme di servizi ad esso correlati: dal supporto online e telefonico per l'installazione, al manuale di documentazione stampato, anch'esso comunque liberamente disponibile su Internet, agli strumenti di configurazione automatica.

La maggior parte delle distribuzioni rimane comunque disponibile su Internet e gratuita; attualmente il numero di distribuzioni presenti si attesta su 382 differenti versioni⁵⁰. Qualsiasi utente, quindi, avrà la possibilità di scegliere il prodotto più adatto alle proprie esigenze. Esistono, infatti, distribuzioni specifiche per diversi target: da quelle ridotte per funzionare anche su hardware molto datato, a quelle create specificatamente per essere utilizzate su macchine firewall, o su filtri antivirus e antispam per i server di posta, da quelle per poter eseguire direttamente anche particolari software Windows a quelle per computer portatili.

2.4.2 Costi di licenza per il software aggiuntivo

Come visto nel paragrafo precedente, il sistema operativo in sé non

⁵⁰ Fonte: www.distrowatch.org (04.02.2005)

permette all'utente di svolgere particolari funzioni, se non un generico utilizzo della macchina, principalmente legate alla gestione di file e directory. E' ormai prassi includere nella distribuzione di un sistema operativo dei semplici programmi per svolgere le operazioni più comuni: nel mondo Microsoft Windows, esempi classici sono costituiti da “Blocco Note”, semplice editor di testo, o da “Microsoft Image Viewer” per la visualizzazione di immagini. Per altri tipi di software, principalmente legati all'utilizzo di Internet, la situazione è diversa: Microsoft Internet Explorer, ad esempio, o il client di posta elettronica Outlook Express, versione semplificata di Outlook di Office, sono sempre stati gratuiti e inclusi nella distribuzione ufficiale di Windows⁵¹. Per poter compiere operazioni più “complesse”, l'utente è così costretto all'utilizzo di altri software e nel caso dei prodotti commerciali, all'acquisto delle rispettive licenze. Gli esempi più rappresentativi sono senza dubbio la suite di produttività personale Microsoft Office, i programmi di fotoritocco (come Jasc Paint Shop Pro o il più celebre Adobe Photoshop) ed i software per la gestione dell'audio.

La spesa necessaria per l'acquisto delle licenze di questi programmi è notevole ed ha una influenza decisiva sul costo totale della messa in opera di ogni personal computer. L'ammontare totale di questa voce di spesa dipende strettamente dalle operazioni che dovranno essere svolte con la

⁵¹ Preferiamo non affrontare l'esempio di Windows Media Player, visualizzatore multimediale per Windows, perché è ancora in atto una causa tra Microsoft e la Commissione Antitrust dell'Unione Europea a riguardo; la prima sentenza ha condannato l'azienda ad un risarcimento di 497 milioni di euro e alla distribuzione di Windows senza l'integrazione del software in questione. Microsoft ha fatto ricorso in appello, ma sarà comunque obbligata all'esclusione del Media Player dal sistema operativo in attesa che il secondo grado di giudizio abbia termine.

postazione in esame e dalla scelta degli applicativi: l'utilizzo di una macchina per la sola navigazione web e posta elettronica necessiterà dell'installazione, e quindi del sostenimento del costo di licenza, del solo sistema operativo, mentre nel caso di configurazioni più "esigenti" la situazione sarà profondamente diversa.

Per postazioni più "versatili", come ad esempio per macchine di tipo "*fat client*"⁵² frequentemente utilizzate in ambito aziendale, riteniamo che l'acquisto di suite di produttività personale, come Microsoft Office, sia assolutamente necessaria per poter svolgere le operazioni più comuni, come la stesura di un documento o la composizione di un foglio di calcolo. Nel caso, invece, di utenze più professionali nel campo del fotoritocco o dell'editing audio-video, si renderà indispensabile anche l'acquisto di ulteriori software. Si può facilmente comprendere come, in caso di utilizzo versatile di una postazione, la spesa da sostenere per le licenze di tutti i programmi possa abbondantemente superare il costo sostenuto per l'acquisto dell'hardware e della licenza del sistema operativo, esponendo pesantemente l'utente al cosiddetto "rischio di aggiornamento" della versione con il conseguente acquisto di nuove licenze.

Il settore del software per Linux negli ultimi anni si è arricchito di prodotti open source, validi e in molti casi assolutamente paragonabili con le controparti commerciali sviluppate dalle più importanti software house.

Nel corso del lavoro abbiamo già affrontato il caso di Mozilla Firefox, il

⁵² Cfr § 3.2.

browser open source che rapidamente sta aumentando la sua diffusione, nonostante il dominio ancora netto di Internet Explorer. Come nel caso del browser della Mozilla Foundation, in questi ultimi tempi sono stati sviluppati da comunità di programmatori dei software che sono assolutamente paragonabili, e molte volte superiori e più apprezzati, dei concorrenti commerciali.

In ambito server, uno degli esempi più rappresentativi è “*Apache web server*”. Software, rilasciato sotto licenza GPL, che svolge la funzione di server web e che ha rapidamente raggiunto una posizione dominante in questo particolare settore⁵³. Secondo dati *Netcraft*, Apache è il server web utilizzato dal 68,83% dei server Internet⁵⁴. Altre esempi di successo sono quelli di “*Samba*”, software e protocollo utilizzato per permettere l'accesso alle risorse condivise in una rete mista Windows Linux, o ancora “*The GIMP*”, *The GNU Image Manipulation Program*, software di editing fotografico e video, utilizzato anche in importanti produzioni cinematografiche digitali⁵⁵. Un caso a parte è poi costituito da OpenOffice, il cui diretto “concorrente” e punto di riferimento è senza dubbio l'ottimo Microsoft Office.

Il prodotto open source, rilasciato sotto licenza GPL⁵⁶, ha rapidamente

53 Nel caso dei server si può creare incomprensione a causa dei termini utilizzati: il “server” può essere inteso sia come *la macchina*, l'hardware, che presta un determinato servizio, come ad esempio quello di accesso web, che come il *software* che permette ciò. Apache è, appunto, un software che assolve la funzione di server web.

54 Fonte: Netcraft, “*Web Server Survey*”, Febbraio 2005, disponibile su news.netcraft.com/archives/web_server_survey.html

55 Queste applicazioni saranno analizzate nel corso dei capitoli successivi. Cfr. § 3.3 e § 4.3.1 e § 4.3.2.

56 Le origini della suite software in questione risalgono al progetto *StarOffice* di Sun

raggiunto la maturità ed ora può vantare un gran numero di funzionalità, un'ottima stabilità e una pressoché completa compatibilità con i formati dei file prodotti dalla suite Microsoft. Quest'ultimo è una caratteristica fondamentale per potersi imporre in un mercato in cui, di fatto, esiste un solo attore⁵⁷.

Tutti i software sopra menzionati sono open source e liberamente disponibili su Internet; qualora quindi dette applicazioni possano soddisfare le esigenze delle postazioni da implementare, risulta chiaramente come il costo sostenuto per l'installazione di questi software sulle postazioni, precedentemente definite “versatili” e “professionali”, sia *nullo*.

2.4.3 Costi per l'acquisizione di hardware

La pronunciata modularità di Linux, così come dei maggiori software ad esso dedicati, permette, come già illustrato, una notevole flessibilità e possibilità di personalizzazione. La flessibilità *a livello kernel*, inoltre, consente la limitazione, o l'estensione, delle funzionalità dello stesso sistema operativo.

In questo modo è possibile utilizzare le ultime versioni del kernel, al passo con i tempi e con le ultime funzionalità, anche su macchine datate, obsolete

Microsystems che decise di sviluppare un prodotto alternativo a Microsoft Office. Nell'estate del 2000 Sun decise di rilasciare il codice del proprio software come open source, dando origine ad una separazione (*forking*) dei rami di sviluppo: nacque il progetto OpenOffice.org, libero e disponibile gratuitamente, e StarOffice, direttamente sviluppato da Sun e venduto ad un prezzo concorrenziale.

⁵⁷ Per un più approfondito confronto tra le due suite di produttività personale, cfr. § 4.1.1.

da anni. Una simile possibilità è molto utile nel caso di postazioni dedicate principalmente ad una particolare funzione come elaborazione testi, accesso ad Internet o firewalling. Nel caso dei sistemi operativi tradizionali, invece, come Microsoft Windows o Apple MacOS X, i miglioramenti a livello kernel, e quindi le funzionalità del sistema, vengono effettuate ed implementate nella versione successiva del prodotto, limitando quindi le vecchie versioni alle originali caratteristiche. Se una specifica versione di Windows non presenta il supporto ad un determinato hardware, o protocollo di comunicazione, per avvalersene sarà necessario aggiornare l'intero sistema operativo, passando alla *release* successiva. Con l'avanzare dello sviluppo di un software, però, crescono proporzionalmente anche i requisiti minimi hardware, rendendo impossibile l'utilizzo delle versioni più recenti su macchine datate. Con Linux questo non avviene: se su una macchina datata, come ad esempio un *Intel Pentium* di prima generazione, viene installata l'ultima versione del kernel Linux, questa sarà in grado di sfruttare le ultime funzionalità in materia di controllo del traffico di rete o condivisione di banda dati permettendone, ad esempio, l'utilizzo come firewall o gateway⁵⁸.

La flessibilità *a livello software* ha orizzonti più ristretti, soffrendo di inevitabili limitazioni dovute alla specifica capacità di calcolo ed elaborazione della postazione. L'utilizzo dell'ultima versione di un

⁵⁸ Utilizzo di una macchina come “via di accesso” (e parallelamente “di uscita”) per la rete Internet o per altri servizi. Sistema spesso utilizzato per accentrare tutti gli accessi su una stessa macchina al fine di controllare i dati trasmessi, effettuare statistiche o filtrarne il contenuto.

wordprocessor su una macchina obsoleta risulterà pressoché impossibile perché il sistema in questione non è in grado di eseguire i calcoli necessari all'esecuzione in un tempo accettabile per l'utente. Queste limitazioni sono, però, legate principalmente ai programmi scritti per essere utilizzati con interfaccia grafica, le cui richieste hardware sono, in tutti i campi, progressivamente aumentate col tempo. I software testuali, basati sulla *CLI (Command Line Interface)*, invece, si dimostrano più parchi nei requisiti hardware e più sfruttabili quindi su macchine obsolete; un'importante frazione di questo tipo di programmi è rappresentata dai server, software utilizzati per prestare servizi all'interno di una rete. Soprattutto all'interno di piccole LAN aziendali è sufficiente una macchina con modeste risorse hardware per prestare servizi come condivisione file, server di stampa, gateway Internet per l'intera rete e filtraggio anti-virus e spam per le postazioni client.

2.4.4 Costi per la manutenzione dei client

Le macchine client costituiscono la maggior parte delle macchine in una infrastruttura di rete e l'operazione di mantenerle continuamente aggiornate ed efficienti è un'importante fonte di spesa ed impiego di tempo ed energie. Le mansioni di chi è responsabile per le macchine si concretizzano solitamente con installazione degli aggiornamenti al software, controllo

dell'integrità hardware, backup periodico dei dati e controllo sull'efficienza dei sistemi.

Quest'ultimo aspetto può essere minacciato da vari fattori, tra i quali ricordiamo la pessima abitudine di molti utenti di installare numerose applicazioni, senza preoccuparsi della rimozione degli stessi, e le temute infezioni da malware. Nel primo caso, una oculata gestione dei permessi⁵⁹ può impedire l'installazione del software da parte dell'utente comune, trasferendo però il compito di farlo sull'amministratore di sistema che dovrà occuparsi di ogni singola macchina. L'alternativa di assegnare i permessi per installare applicazioni, però, è uno scenario che preoccupa qualsiasi amministratore di sistema che lavori con Windows: spesso gli utenti aggiungono "componenti" al sistema operativo senza preoccuparsi delle conseguenze.

Uno dei difetti più criticati dei sistemi Windows, inoltre, è quello di "appesantirsi", diventando sempre più lento, con il tempo. Spesso legato ad un problema di riempimento delle chiavi di registro dello stesso sistema operativo, rende necessaria una nuova installazione dell'intero sistema operativo e dei software aggiuntivi utilizzati. Compiere un'operazione del genere, quantificabile in un lavoro di qualche ora⁶⁰, su una moltitudine di macchine costituisce una inaccettabile perdita di tempo e denaro.

⁵⁹ Si rimanda al § 2.1.3.

⁶⁰ Considerando 90 minuti di durata media per una installazione di Windows XP Home, 40 minuti per l'intero pacchetto Microsoft Office e circa 60 minuti per l'installazione di software aggiuntivi, calcolando anche i frequenti, e necessari, riavvii richiesti dal sistema durante le suddette operazioni. Dati raccolti dall'autore.

Il secondo ordine di problemi è spesso costituito dall'infezione da worm e virus; anche in questo caso si rende necessario un intervento sulla macchina, utilizzando strumenti appositi per la rimozione di questo particolare tipo di software. In taluni casi può essere anche necessaria la completa reinstallazione del software, ivi compreso il sistema operativo.

I sistemi Linux non soffrono di questo particolare problema di “appesantimento col tempo”: non esiste nessun “Registro di Sistema” che si riempie col passare del tempo e l'installazione del software viene eseguita in maniera assolutamente trasparente e, quindi in caso di rimozione, sono facilmente annullabili tutte le modifiche apportate al sistema. Un'eventuale sovrabbondanza di programmi installati, inoltre, non può appesantire l'intero sistema, dato che i servizi ed i software da eseguire all'avvio (“boot”) della macchina sono di competenza dell'utente amministratore e solo da questi modificabili. L'unico ambiente che quindi potrebbe soffrire dell' “ingenuità” dell'utente comune sarebbe quello dell'utente stesso e non dell'intero sistema. Qualora divenisse comunque necessario un intervento dell'amministratore di sistema giocherebbero un ruolo fondamentale, in termini di risorse impiegate, l'accesso da remoto e lo shell scripting, caratteristiche già affrontate nel paragrafo 2.3. Con semplici operazioni e comandi, l'amministratore avrebbe accesso alle macchine remote, potrebbe risolvere il problema (rimuovendo software o eseguendo le altre operazioni necessarie) e ripristinare il normale funzionamento della postazione.

Potrebbero essere conseguiti vantaggi analoghi anche nell'esecuzione di procedure periodiche come l'aggiornamento del sistema e il backup dei dati: uno shell script può essere composto in pochi secondi ed eseguito ad intervalli regolari.

In un recente intervento presso l'*Open Source Development Lab (OSDL)*, Craig Manning, IT Manager di Cisco USA, ha affermato che un amministratore di sistema può gestire circa 40 postazioni Windows; con sistemi Linux il rapporto sale ad un numero che varia dalle 200 alle 400 macchine. Manning è attualmente responsabile per il processo di migrazione a Linux dei sistemi desktop della multinazionale americana.

2.5 Vendor tie-in

Utilizzare software proprietario pone il cliente, l'azienda nel nostro caso, in una situazione di notevole dipendenza dalla casa sviluppatrice, soprattutto quando l'applicazione interessata è "*business critical*".

Utilizzare software proprietario, prodotto dal leader nel settore e pressoché unico attore nel mercato, rende il cliente, l'azienda, troppo dipendente ed ininfluente per poter esercitare una qualsiasi forma di contrattazione con il fornitore.

Affrontiamo ad esempio il caso di un software "gestionale" di tipo proprietario. Una scelta del genere colloca il cliente-azienda in una

posizione di dipendenza dal fornitore a nostro avviso molto rischiosa: l'adozione di una applicazione di questo tipo comporta una dipendenza dallo strumento *specifico* che aumenta con il passare del tempo e con l'utilizzo intenso. Il software gestionale è di norma utilizzato con il supporto di un *database* nel quale vengono memorizzate tutte le informazioni da gestire. Le aziende sviluppatrici fanno uso di una propria tecnica per l'*encoding*⁶¹ dei dati, rendendo questi ultimi sostanzialmente inutilizzabili da altri software gestionali. Non è importante sapere che i dati siano memorizzati, ma sapere *come* siano memorizzati. L'unica tecnica che permette, a chi non conosca esattamente come avviene la codifica del file, di sviluppare, seppur con alterni ed incerti risultati, applicazioni compatibili con lo specifico formato è il *reverse engineering*, ma non è possibile garantire in nessun modo il corretto funzionamento e la completa compatibilità. In questo caso l'utilizzo di tecnologie proprietarie rende potenzialmente impossibile la coesistenza di diverse applicazioni che utilizzino il medesimo encoding.

Quanto successo con i file usati dalla suite Microsoft Office e la compatibilità con Openoffice è stato molto indicativo: gli sviluppatori della suite libera hanno necessitato di molto tempo al fine di comprendere come l'encoding dei file *xls* di Excel o *doc* di Word funzionasse. Attualmente sono

⁶¹ Tipo di codifica utilizzata per immagazzinare dati. L'esempio più comune è l'encoding dei file, normalmente definita "tipo del file". Un file audio *mp3* usa un tipo di encoding noto a tutti, di pubblico dominio; per queste ragioni, un file di questo tipo può essere utilizzato, e sfruttato, da un largo numero di applicazioni. Caso analogo per i file di immagini *jpg*. Non si può dire altrettanto, ad esempio, per il file di immagine *tiff*, tecnologia proprietaria di Adobe, leader nel mercato della grafica e del foto ritocco con Photoshop.

riusciti nell'intento di garantire la compatibilità, ma, teoricamente, sarebbe sufficiente un importante cambiamento nella tecnologia di Microsoft per portare il supporto dei formati in Openoffice ad un livello assolutamente insoddisfacente. La decisione dell'azienda di Redmond di pubblicare le specifiche dei propri formati dei file Office⁶² è molto recente e ha creato un importante presupposto per la compatibilità futura di questi applicativi.

Nel mondo dei software gestionali la situazione è analoga, ma, attualmente, nessuna azienda sviluppatrice ha deciso di seguire la scelta di Microsoft e pubblicare le proprie specifiche per l'encoding; ricordiamo che un'operazione di questo tipo non “scopre le carte” del software completo, ma illustra solo come l'applicazione memorizzi i dati, garantendo la portabilità su altri prodotti.

Aumentando, con il tempo e con l'utilizzo, la mole di dati amministrati dal software gestionale, le risorse necessarie ad effettuare un cambiamento di applicazione diventano sempre maggiori e senza nessuna garanzia di successo. Qualora, infatti, si dovesse procedere alla sostituzione dell'applicazione con un altro prodotto, si dovrà necessariamente scegliere un'alternativa compatibile, se presente, ovvero provvedere alla conversione dei dati che può diventare fonte di difficoltà ed impiego notevole di risorse.

La situazione è analoga nel caso del sistema operativo utilizzato o di altre applicazioni; è possibile che i problemi non vertano sulla codifica dei file, ma semplicemente sulla piattaforma utilizzata. La necessità di procedere

⁶² Fonte: <http://www.microsoft.com/office/xml/default.mspx>

agli aggiornamenti del sistema (hardware e software) non quando ritenuto necessario, ma quando l'azienda sviluppatrice lo ritenga opportuno, pone il cliente in una condizione di forte ed inaccettabile dipendenza.

Un esempio recente di “rischio di aggiornamento” è stato rappresentato dall'interruzione da parte di Microsoft del supporto alle più vecchie versioni di Windows; un qualsiasi utente di Windows 98, ad esempio, aveva dinanzi l'alternativa di lavorare con un prodotto non aggiornato, e non aggiornabile, ovvero di acquistare una licenza per la nuova versione del sistema operativo. La comunicazione recente di Microsoft⁶³ in merito allo sviluppo di Internet Explorer 7, nuova versione del browser web, rappresenta un caso analogo: il software sarà compatibile solo con Windows XP. Data la scelta dello sviluppatore di escludere dal supporto anche il molto diffuso Windows 2000, gli utenti saranno costretti a scegliere tra l'aggiornamento dei sistemi o la rinuncia alle funzionalità del nuovo browser.

Una soluzione per non correre simili rischi è, appunto, quello di utilizzare software open source che non genera alcun effetto di *vendor tie-in*; il codice sorgente delle applicazioni è disponibile per l'analisi e lo studio. Gli aggiornamenti del kernel, cuore del sistema operativo e potenziale “fonte” di bug o falle di sicurezza, viene costantemente aggiornato e reso disponibile in Rete. Nel caso, quindi, di sostituzione di un software di questo tipo, la conversione dei dati non sarà una procedura empirica e rischiosa, ma analizzabile, pianificabile e *realizzabile*.

⁶³ Fonte: <http://blogs.msdn.com/ie/archive/2005/02/28/382054.aspx>

Alla luce di quanto esposto in questo paragrafo, riteniamo che il semplice fatto che Linux sia open source possa essere una valida, ancorché non unica, motivazione per migrare.

3. Analisi e pianificazione preliminare

Nel corso di questo capitolo descriveremo le operazioni, gli strumenti e gli approcci più efficienti per poter procedere con l'implementazione di Linux nella propria realtà aziendale e trarne il massimo vantaggio.

Inevitabilmente la migrazione verso Linux e l'abbandono di sistemi basati su piattaforma Microsoft Windows, e sui prodotti ad esso correlati, creerà difficoltà iniziali, rallentamenti e necessità di adattamento. Sarà a nostro avviso necessario procedere con delle attività preliminari di indagine e pianificazione al fine di prevedere, nei limiti del possibile, ciò che si potrà incontrare durante la migrazione e poter intervenire tempestivamente. Come qualsiasi processo di innovazione e di radicale cambiamento, che interessi diversi soggetti con competenze e ruoli diversi, la migrazione dei sistemi può rivelarsi un fallimento, qualora non siano stati analizzati e pianificati nei minimi dettagli i passi da compiere e le soluzioni da adottare.

Nel capitolo precedente abbiamo illustrato le motivazioni principali che dovrebbero condurre all'implementazione di Linux in azienda; nel presente capitolo, invece, affronteremo le attività di pianificazione precedenti la migrazione e, successivamente, la descrizione degli scenari all'interno dei quali può non essere consigliabile procedere con l'implementazione di Linux. Concluderemo il capitolo con una nostra proposta per una possibile configurazione di network aziendale, illustrandone i vantaggi principali.

3.1 Analisi preliminare

La migrazione verso piattaforma Linux può comportare importanti vantaggi all'interno dell'azienda, una maggior produttività, riduzione dei costi e flessibilità.

Come verrà illustrato, è necessaria un'attenta e preliminare procedura di analisi e pianificazione al fine di valutare opportunamente i fattori di successo e di insuccesso. L'amministrazione interessata sarà così tenuta a compiere valutazioni di questo tipo e verificare gli opportuni vantaggi che potrà trarne. In particolari configurazioni, a nostro avviso, sussistono delle difficoltà e barriere tali da poter rendere la migrazione eccessivamente onerosa ed inefficiente. Conoscere preventivamente i rischi e saper affrontare, preparati, tali ostacoli può rappresentare la differenza tra il successo ed il fallimento.

Le realtà aziendali possono differire profondamente l'una dalle altre: le configurazioni di un network, le singole postazioni ad uso degli impiegati, le loro attività, l'utilizzo della Rete e delle Intranet, rendono ogni scenario particolare e differente da altri. Non esiste un'*unica* "ricetta" per poter implementare con successo Linux nel proprio network aziendale. E' necessario, quindi, svolgere un'analisi sui *propri* sistemi, sulle esigenze e caratteristiche della propria infrastruttura e delle proprie postazioni.

La prima distinzione da eseguire è quella tra macchine client e macchine server. Gli ostacoli incontrabili con queste ultime sono di norma più semplici da superare alla luce di diverse motivazioni: in primo luogo, sono macchine destinate a prestare servizi e che non prevedono un'interazione diretta tra macchina e utente finale. La configurazione, manutenzione e utilizzo del server sarà di competenza dell'amministratore di sistema, creando così in caso di migrazione dei problemi di adattamento relativi ad un ristrettissimo numero di individui; questi, inoltre, sono normalmente preparati tecnicamente e possono considerare la migrazione come una "sfida", assumendo l'atteggiamento corretto per superare gli inevitabili ostacoli che la novità porterebbe.

La sfida più importante, e difficile da superare, è quella della migrazione dei client, ovvero delle postazioni dei singoli utenti; il nostro lavoro affronterà principalmente questo aspetto.

3.2 Distinzione in thin e fat client

La prima domanda alla quale rispondere è relativa alla tipologia di macchine client utilizzate all'interno della propria struttura: *thin client* o *thick* e *fat client*. Questo tipo di distinzione è fondamentale per poter determinare le difficoltà del processo di migrazione e le relative contromisure da adottare.

Nel caso dei *thin client*, le postazioni vengono utilizzate come una sorta di terminali al fine di accedere a risorse di rete installate su un server, dal quale sono strettamente dipendenti. Un esempio molto comune nelle realtà aziendali è l'utilizzo di terminali *thin* come interfaccia al server web, o al database server, dell'azienda; in questi casi, la postazione utilizzata per accedere alle risorse di rete è pressoché ininfluenza nelle prestazioni e nella produttività del lavoro svolto con essa, mentre giocano dei ruoli fondamentali la velocità del collegamento di rete e la potenza computazionale del server centrale.

I *fat client*, e analogamente i *thick client*, invece, sono personal computer “autosufficienti” nel proprio lavoro e che, al fine di compiere le diverse operazioni, fanno ricorso alle proprie risorse locali. Normalmente, l'accesso da parte di queste postazioni a risorse centrali si limita all'autenticazione all'interno di un dominio o la memorizzazione dei dati su server centrali. In questo caso, quindi, l'utente può svolgere le proprie mansioni anche in un contesto avulso dalla rete aziendale (*offline*), utilizzando software e dati presenti nel proprio personal computer.

3.2.1 Thin Client

La migrazione a Linux di postazioni *thin client* si rivelerà nella maggior parte dei casi semplice e sostanzialmente *neutra* nei confronti dell'utente.

Una volta identificata l'interfaccia di connessione utilizzata per accedere al server, sarà sufficiente verificare la presenza dello stesso applicativo per Linux o la disponibilità di strumenti analoghi. Un esempio potrebbe essere costituito da una postazione caratterizzata in questo modo:

- utilizzo di browser web per accedere a portale o Intranet aziendale
- utilizzo di applicazioni web aziendali
- non utilizzo di software di produttività personale installato localmente
- utilizzo elementare della posta elettronica tramite interfaccia web
- accesso a risorse condivise di file e stampanti

Una configurazione di questo tipo non crea, in sede di migrazione, particolari difficoltà perché non fa utilizzo di particolari applicazioni o funzioni: risulta infatti in maniera chiara come il browser internet, Internet Explorer o Mozilla Firefox, sia lo strumento principale utilizzato da una simile postazione, sia per l'accesso al portale aziendale, o alla Intranet, che per la posta elettronica.

In sistemi Windows questo strumento sarà rappresentato da Internet Explorer nella maggior parte dei casi; abbiamo già illustrato nel paragrafo 2.1.5 come Mozilla Firefox abbia ricevuto giudizi favorevoli e raggiunto

importanti traguardi in brevissimo tempo. Ai fini della migrazione si potrebbe installare in un primo momento il software in questione sulle postazioni *thin*, senza però modificare il sistema operativo. Dopo un periodo di prova, si potrà compiere il secondo passo, sostituendo il sistema operativo Windows con Linux. Questo passaggio verrà illustrato successivamente.

3.2.2 Fat Client

Nel caso dei *fat e thick client* il processo di migrazione diviene estremamente più complesso rispetto a quanto affrontato con i terminali *thin*. Le maggiori difficoltà sono principalmente relative al numero di applicazioni utilizzate nelle postazioni e alle funzionalità di queste. Gran parte del nostro lavoro si concentrerà proprio su questo tipo di postazioni che rappresentano la sfida più difficile, e spesso più frequente, da superare nel corso della migrazione a piattaforma Linux ed al software open source.

Il primo passo, a nostro avviso, consta nel determinare il ruolo svolto dal client e le applicazioni utilizzate più frequentemente nelle diverse postazioni. Le informazioni raccolte saranno di importanza fondamentale per poter pianificare accuratamente la propria migrazione, per non commettere errori comuni e facilmente evitabili ed, in definitiva, per avere successo nell'operazione.

La maggior parte di questo lavoro preliminare può essere svolta tramite osservazione diretta ovvero intervista e questionari da presentare agli utenti che, dopo la migrazione, dovranno *effettivamente* utilizzare le “nuove” postazioni.

3.3 Ruolo del client ed applicazioni utilizzate

Inizialmente è opportuno provvedere alla classificazione dei client, raggruppandoli in categorie omogenee di postazioni.

Il primo punto da affrontare è la raccolta delle informazioni sulle applicazioni utilizzate, sulla frequenza e sulle funzionalità utilizzate. Una classificazione pratica, frequente in un'ottica aziendale, potrebbe essere simile a quanto illustrato nella Tabella 3, dove si propone una rappresentazione dello scenario e una prima classificazione in applicazioni e loro utilizzo da parte di entrambe le categorie di client.

	<i>Basic Client</i>	<i>Interm / Advanced Client</i>
<i>Applicazioni Internet</i>	Browser Web	Browser Web
	Posta Elettronica	Posta Elettronica (avanzato)
<i>Applicazioni Office</i>	Instant Messaging	Instant Messaging
	Wordprocessing	Wordprocessing (avanzato)
	Fogli di calcolo	Fogli di calcolo (avanzato)
<i>Altre Applicazioni</i>	Presentazioni	Presentazioni
	-	Foto ritocco e grafica
		Publishing
		Altri software (CAD, framework di sviluppo)
<i>Utilizzo Infrastruttura Locale</i>	Accesso filesystem condivisi Accesso stampanti di rete Accesso a database server Accesso a groupware aziendale	

Tabella 3: Classificazione dei client

Come è possibile osservare in Tabella 3, entrambi i client fanno utilizzo delle medesime risorse dell'infrastruttura locale, sinteticamente indicate nell'accesso a file e directory condivise via network, stampanti di rete, database server e software groupware o di altro tipo che prevedano un accesso o interazione con macchine remote. I primi tre punti, invece, mostrano l'utilizzo di specifiche applicazioni e le funzionalità necessarie per garantire produttività a queste postazioni. Notiamo come gli “Intermediate / Advanced Client” presentino non solo una maggiore ricchezza di software utilizzato, ma anche un diverso sfruttamento di talune applicazioni presenti anche nei “Basic Client”.

Le difficoltà di una migrazione verso sistemi Linux sono a nostro avviso riconducibili principalmente, se non *esclusivamente*, al numero di

applicazioni specifiche da implementare con il nuovo sistema operativo: per ogni software, infatti, è necessario analizzare che tipo di utilizzo ne viene fatto (elementare, intermedio o avanzato), identificare l'alternativa software da utilizzare dopo la migrazione a piattaforma Linux e risolvere eventuali difficoltà e problemi relativi al cambiamento. Un maggior numero di applicazioni, quindi, implica un ulteriore studio di fattibilità e l'insorgere di possibili complicazioni e difficoltà nell'adattamento da parte del singolo utente che devono essere risolte al più presto al fine di non minare la produttività delle singole postazioni.

3.4 Attuali problemi ed esigenze delle postazioni

Con riferimento a quanto sopra descritto, il momento dell'analisi preliminare costituisce a nostro avviso un'ottima occasione per verificare quali siano i problemi delle singole postazioni (o categorie di client) e del network nel suo complesso.

L'indagine iniziale non deve quindi limitarsi alla descrizione di uno stato “*as is*”, ma andare oltre ed identificare quale dovrebbe essere lo scenario “*to be*”. Una volta individuati questi aspetti è opportuno verificare come questi possano essere risolti in sede di implementazione di Linux nella nostra infrastruttura e pianificare la loro risoluzione coordinatamente al processo di migrazione.

Qualora determinate postazioni utilizzino software solo saltuariamente, o ne facciano un uso elementare, è a nostro avviso necessario pianificare strategie alternative di sostituzione. Fare ricorso a software libero, in questo caso, permetterebbe una notevole riduzione dei costi di licenza evitando i paradossi legati alle più diffuse condizioni di licenza: acquistare diverse licenze di software professionale, da utilizzare su diverse postazioni saltuariamente, costituisce un notevole spreco di risorse. L'analisi preliminare alla migrazione da noi proposta può quindi costituire un'importante occasione per verificare l'esistenza di eventuali disfunzioni ed inefficienze ed il loro impatto sulle attività e per pianificare interventi volti ad eliminare simili evenienze.

3.5 Quando non migrare

Come visto precedentemente, e come meglio illustreremo in seguito, l'implementazione di Linux in azienda, sostituendo Windows, non pone problemi particolari dal punto di vista del sistema operativo e dell'infrastruttura. La barriera più difficile da superare è creata dall'eventuale incompatibilità di particolari applicazioni che, qualora siano critiche per le attività dell'azienda, pongono ostacoli alla migrazione e rendono necessaria un'attenta pianificazione ed il disegno di una strategia *ad hoc* per superarli. Linux è un sistema in continua evoluzione e presenta

lati negativi e difetti come tutte le altre piattaforme; Windows, Apple MacOS presentano lo stesso scenario: non esiste un prodotto perfetto. E' quindi necessario valutare la convenienza ed il vantaggio apportato da una migrazione dell'intero network aziendale.

Particolari scenari possono presentare difficoltà maggiori di altri e rendere troppo onerosa, in termini di risorse, la migrazione; in taluni casi, quindi, il mantenimento della piattaforma Windows preesistente, per quanto caratterizzata da difetti, oneri e rischi particolari, esposti in precedenza, potrebbe costituire la soluzione più conveniente, almeno sul breve termine. Riteniamo, infatti, che l'adozione di tecnologie proprietarie, ed in particolare quelle riconducibili ad un monopolista, possa creare, se non l'ha già fatto, situazioni di squilibrio che dovrebbero comunque essere risolte. Uno scenario di questo tipo può essere relativo allo stesso modo al sistema operativo, alla tecnologia per la memorizzazione dei dati in database o all'utilizzo di un particolare software gestionale.

In questo paragrafo illustreremo gli scenari più frequenti che possono rendere l'implementazione di Linux troppo onerosa e potenzialmente non conveniente; nella maggior parte dei casi, come visto, gli ostacoli maggiori sono relativi proprio alla impossibilità di migrare determinate applicazioni o infrastrutture. A nostro avviso le configurazioni più problematiche sono:

1. preponderante diffusione dei *fat* e *thick client*

2. utilizzo frequente e diffuso di applicazioni non migrabili
3. recente processo di aggiornamento dei sistemi

I punti 1. e 2. sono strettamente legati e rappresentano uno scenario nel quale la maggior parte delle postazioni fanno continuo uso di risorse locali, in termini di dati e di applicazioni eseguite; le operazioni di rete più frequentemente utilizzate da questo tipo di postazioni sono la memorizzazione, o lettura, di file contenuti in macchine remote, l'utilizzo di stampanti di rete, la condivisione di file con altre postazioni e il saltuario utilizzo di interfacce di controllo per software via network. Una configurazione del genere presenta quindi la maggior parte delle applicazioni installate su ogni singola postazione ed eseguite localmente.

Come affrontato in precedenza, migrare verso piattaforma Linux implica anche migrare le diverse applicazioni utilizzate; ai fini della migrazione, sarà quindi necessario un attento studio di pianificazione e analisi delle alternative, in modo tale da evitare rallentamenti nel normale svolgimento delle attività delle postazioni. Aumentando il numero di applicazioni alternative che dovranno essere eseguite dalle postazioni, potranno aumentare anche le difficoltà incontrate dagli utenti nel prendere dimestichezza con i nuovi strumenti¹. Analogamente, un numero alto di applicazioni aumenta inevitabilmente le probabilità di incontrare problemi di incompatibilità o implementabilità sotto Linux, creando quindi difficoltà

¹ Cfr. § 4.5.1 e § 4.5.2 relative al “Fattore Umano” e alla formazione degli utenti.

nel processo di migrazione.

Nel capitolo successivo illustreremo alcune strategie utilizzabili al fine di risolvere taluni problemi di incompatibilità, ma è possibile che talvolta non possano essere risolti. In questi casi, l'unica possibilità è rappresentata da mantenere Windows, seppure in diverse configurazioni: nelle singole postazioni *client* che lo necessitano, o in un server centrale che renderà disponibili le applicazioni all'intera rete oppure utilizzando il sistema operativo Microsoft con VMWare e software analoghi². Qualora il numero di postazioni che dovranno mantenere Windows sia alto e, soprattutto, qualora il numero di applicazioni incompatibili sia rilevante, riteniamo che il processo di migrazione si complichino eccessivamente, rendendo l'operazione estremamente complessa e soggetta al pesante ed oneroso rischio di insuccesso.

Quanto sopra, ovviamente, in un'analisi di tipo “*as is*”. Come affrontato precedentemente in merito all'incidenza dei costi di licenza e delle tecnologie proprietarie in generale, potrebbe essere colta l'occasione, durante l'analisi e pianificazione preliminare, per valutare quale dovrebbe essere lo scenario “*to be*”. A riguardo proporrò delle soluzioni nel corso dei paragrafi successivi.

Un'altra circostanza che potrebbe sconsigliare la migrazione è quella indicata precedentemente al punto 3. : se i sistemi e l'infrastruttura in generale sono stati recentemente aggiornati, la prospettiva di “cambiare”

² Cfr. a riguardo § 4.1.3 e § 4.1.4.

potrebbe non essere la migliore soluzione. L'aggiornamento cui ci riferiamo è principalmente legato al software e alle tecnologie di rete e di infrastruttura utilizzate: la sostituzione di versioni datate di Windows con il recente XP, ad esempio, costituirebbero un importante capitolo di spesa sostenuto per l'acquisto delle licenze delle diverse postazioni. Allo stesso modo, il rinnovamento del software aggiuntivo utilizzato sui client rappresenterà molto probabilmente un notevole utilizzo di risorse.

Riferendoci, invece, alle macchine server e alla infrastruttura in generale, illustriamo come ulteriore esempio l'installazione di server basati su prodotti Microsoft come Windows Server 2003, Microsoft Exchange Server o IIS Web Server; il tempo necessario ad implementare e configurare sistemi di rete, soprattutto all'interno di infrastrutture notevoli, è notevole ed ingenti saranno i costi da sostenere per realizzarli.

Alla luce di quanto esposto risulta chiaro che in talune situazioni non sia consigliabile effettuare la migrazione dei sistemi o, quantomeno, sia da valutare molto attentamente prima di procedere. In scenari di questo tipo riteniamo che il momento più opportuno per implementare la migrazione a piattaforma Linux sia quello di un nuovo aggiornamento dei sistemi e del software utilizzato.

3.6 Utilizzo di applicazioni web based

In questo paragrafo illustreremo un'analisi di tipo “*to be*”, tramite la quale proponiamo una possibile configurazione per network aziendali al fine di conseguire notevoli vantaggi in termini di costi di licenza, sicurezza e produttività. Il nostro progetto presuppone un approccio particolare al concetto di network, personal computer e utilizzo delle applicazioni. Riteniamo, infatti, che il massiccio utilizzo di *fat client* all'interno di una realtà aziendale complessa e dotata di numerose postazioni, possa costituire un notevole spreco di risorse per l'acquisizione delle licenze del software, un potenziale rischio per la sicurezza informatica della rete e un'amplificazione del *vendor tie-in*.

Vediamo singolarmente questi punti.

3.6.1 Costi di licenza: scenario attuale e rischi per il futuro

Lo spreco di risorse è imputabile sostanzialmente alla necessità di rendere ogni singola postazione autosufficiente nello svolgimento di una notevole quantità di operazioni ed attività; con questo assetto, sarà necessario provvedere all'installazione di diverse applicazioni su ogni sistema.

Dovendo far ricorso quasi esclusivamente alle proprie “risorse locali”, infatti, il client dovrà essere necessariamente dotato di tutte le applicazioni *potenzialmente* utili nello svolgimento dell'attività caratteristica della postazione in esame. Ad un'impostazione di questo tipo consegue un

importante capitolo di spesa relativo all'acquisto delle licenze per le diverse applicazioni, anche se utilizzate raramente. Queste considerazioni sono naturalmente applicabili solo a software commerciale che preveda, quindi, dei costi di licenza; sarebbero altresì escluse ipotetiche licenze d'uso del software con tecniche di imputazione diverse da quelle normalmente utilizzate (come, ad esempio, una licenza d'uso da pagare in base all'utilizzo *in contemporanea* di diverse postazioni)³. Allo stesso modo, è escluso *de facto* tutto il settore del software libero che, nella maggior parte dei casi, non prevede alcun costo di licenza.

In questa sede vogliamo aggiungere una seconda considerazione sulla quale, a nostro avviso, bisognerebbe riflettere profondamente prima di intraprendere nuovi aggiornamenti di software proprietario di tipo commerciale. Informazioni recenti mostrano che le società sviluppatrici di software sono al lavoro per determinare nuove tecniche di imputazione delle licenze d'uso dei propri programmi, “più adeguate” alle attuali condizioni di mercato e di utilizzo. Fino a questo momento, infatti, era necessario acquistare una licenza per ogni macchina, identificando quindi l'hardware di una postazione come il punto di riferimento per imputare ogni licenza. Recenti tecnologie, però, stanno rapidamente cambiando lo scenario, rendendo possibile l'utilizzo di una singola postazione in multiutenza in grado di eseguire diverse applicazioni contemporaneamente.

³ Naturalmente una simile evenienza risolverebbe solo il problema dei costi di licenza per applicazioni utilizzate saltuariamente. L'Autore non è comunque a conoscenza di licenze di questo tipo. Approfondiamo il tema dell'imputazione dei costi di licenza per singole postazioni in un network all'interno del § 4.1.4.

Ci riferiamo, in primo luogo, alla virtualizzazione dei processi e delle attività, tecnologia resa disponibile negli ultimi tempi e che sta rivoluzionando, ad esempio, i servizi dei provider che operano tramite Internet, offrendo soluzioni di *housing* o *hosting*⁴. Un esempio è costituito da una funzionalità di *Apache Web Server* e di alcuni software analoghi: il *Virtual Host*. Con questa tecnologia un singolo web server è in grado di ospitare, ed eseguire, diversi servizi web, ognuno corrispondente ad un diverso *host* e indirizzo web. La limitazione nel numero è sostanzialmente definita dalle caratteristiche hardware e di elaborazione della macchina. La funzionalità di *Virtual Host* non è avvertibile dall'utente, che non si accorge di contattare la stessa macchina per diversi servizi e siti internet.

Ciò che fino a poco tempo fa era possibile eseguire solo con più macchine ora è sostenibile con un solo server; dal punto di vista delle licenze, escludendo ovviamente il caso di Apache e di tutto il software libero, si è creato un “danno” per le aziende produttrici di software del genere che, per le attuali condizioni di licenza, potrebbero richiedere ai propri clienti solo un acquisto per ogni singola macchina, indipendentemente dal numero di siti web “ospitati”.

Un'altra tecnologia che ha causato un ripensamento dei produttori sui termini della licenza dei propri software è stato quello dei processori

4 Per *hosting* si intende un servizio prestato da un provider, atto ad ospitare dati ed informazioni del cliente su uno spazio web disponibile sulle macchine del provider stesso. Frequentemente viene utilizzato per la creazione di siti web. L' *housing* è un servizio più avanzato: l'azienda infatti “ospita”, da qui il termine, una macchina del cliente, sul quale quest'ultimo ha il pieno controllo.

multicore. Il campo di applicazione, in questo caso, è strettamente hardware e non si riferisce, se non indirettamente, ai software eseguiti dalla macchina. Fino ad oggi, infatti, la maggior parte dei processori legati al mercato non professionale dell'elaborazione dati presentava un singolo *core*, cuore della CPU, destinato alle procedure di calcolo. In tempi recenti, invece, i produttori di processori per il mercato *consumer*, *Intel* e *Amd* in testa, hanno sviluppato CPU *multicore* ed in grado di moltiplicare la capacità di elaborazione dati. In realtà, dal punto di vista strettamente tecnico, il fattore di moltiplicazione non è proporzionale: un processore *dual core* non avrà prestazioni pari al doppio di una CPU analoga *single core*.

Ciò nonostante, alcuni sviluppatori di software, come *IBM*, *Sybase* ed *Oracle*, sono dell'opinione che ad un tale aumento della capacità di calcolo delle macchine debba anche conseguire un corrispondente aumento del costo di licenza dei software. In accordo con quanto affermato da Gartner⁵, cambiamenti di questo tipo nelle licenze d'uso possono portare ad un raddoppio dei costi sostenuti per questo capitolo di spesa entro la fine del 2005. Microsoft, invece, ha affermato che l'adozione di tecnologie *multi core* nei processori non comporterà una moltiplicazione del prezzo delle licenze dei propri software.

Lo scenario futuro non si presenta a nostro avviso così rassicurante da non

5 Fonte: www.gartner.com . Ricerca n. G00123954 “Hardware Improvements Drive Up Software Costs, Force Price Changes”, 15 Ottobre 2004 e n.G00124491 “Microsoft Stance on Multicore Licensing Will Help Customers”, 20 Ottobre 2004.

prendere in seria considerazione le conseguenze di un intenso utilizzo di software commerciale nella propria infrastruttura; l'adozione, infatti, di simili applicativi, soprattutto nel caso di un singolo “attore” nel mercato, crea una pericolosa dipendenza da un singolo fornitore. In aggiunta a ciò, la preponderanza di *fat client* in una realtà aziendale rischia di agire da moltiplicatore degli effetti che possono conseguire ai cambiamenti nelle licenze d'uso.

3.6.2 Fat client e sicurezza

L'installazione di numerosi software su ogni singola macchina aumenta, inevitabilmente, il numero di vulnerabilità cui le stesse sono soggette, assoggettandole quindi ad un maggior rischio di compromissione. Allo stesso modo, una configurazione simile crea anche serie difficoltà nella gestione, manutenzione e messa in sicurezza delle macchine da parte del servizio tecnico; l'aggiornamento di decine di software su ogni postazione facente parte un network aziendale presuppone un notevole dispendio in termini di tempo e risorse, cui spesso consegue un non tempestivo intervento di aggiornamento dei sistemi. Abbiamo visto come il *mix* di sistemi non aggiornati e connettività Internet costituisca un serio problema per la sicurezza di una rete.

La gestione a livello centrale di una moltitudine di postazioni di tipo *thin*

client permette all'amministrazione tecnica di implementare specifiche strategie di controllo e gestione del flusso di dati; identificando la connettività Internet come la causa principale dei danni relativi alla sicurezza informatica, in un network composto da *thin client* sarà facilmente possibile operare un filtro sulle comunicazioni via rete, al fine di identificare e bloccare particolari tipi di dati, frequentemente riconducibili all'infezione dei sistemi da parte di *malware*.

Quanto per ora affermato, non tiene in considerazione il fattore umano, la non “preparazione tecnica” degli utenti e i rischi dovuti all'ingenuità, all'errore e alla “ingegneria sociale” in generale; questi sono elementi difficilmente controllabili a monte e in remoto da parte del servizio tecnico che può limitarsi, salvo particolari accorgimenti, ad intervenire quando il danno, più o meno ingente, è già occorso.

3.6.3 I software groupware

La nostra proposta per un potenziale scenario “*to be*” è quella di fare uso di tecnologie e applicazioni web che rendano possibile e altamente produttivo l'utilizzo di *thin client*. Con strumenti di questo tipo, le singole postazioni vengono private della maggior parte delle applicazioni e sfruttate principalmente come terminali per accedere a risorse centrali. Queste ultime sono rappresentate da server che eseguono applicazioni web, accessibili

tramite browser, rendendole disponibili a tutte le postazioni. Le attività dei client, quindi, si limiterebbero all'utilizzo di un browser web, Internet Explorer o Mozilla Firefox ad esempio, per accedere al “sito” del server centrale, all'autenticazione tramite nome utente e *password* e utilizzo delle applicazioni rese disponibili.

In questo contesto si inserisce una particolare categoria di software che ha conosciuto recentemente una rapida diffusione: i groupware.

Le applicazioni web di questo tipo, come *PhpGroupware*, *eGroupware*⁶ o *Tikiwiki*⁷, hanno una struttura fortemente modulare, permettendo una notevole personalizzazione e possibilità di espandere in futuro le funzionalità, tramite l'installazione di moduli aggiuntivi. Attualmente questi software, personalmente provati dall'Autore, presentano moduli per le seguenti funzionalità, particolarmente utili in ambito aziendale:

- utilizzo della posta elettronica (account multipli)
- gestione di una agenda condivisa
- forum
- gestione per le comunicazioni rapide con il personale e con collaboratori
- gestione di una rubrica condivisa

⁶ Cfr. rispettivamente www.phpgroupware.org e www.egroupware.org. I due software in questione presentano un codice molto simile a causa della divisione, recentemente avvenuta, nei rami di sviluppo. Entrambe le applicazioni sono rilasciate con licenza GPL e liberamente disponibili.

⁷ Cfr. www.tikiwiki.org. Anche questa applicazione è software libero, rilasciato con licenza GPL e liberamente disponibile.

- *filemanager* per accesso a file e cartelle di rete condivise
- *ftp server* per trasferimento generico di file
- client per *instant messaging*
- gestione progetti
- strumenti per la collaborazione nella stesura di documenti (*wiki*)

Dalla lista delle applicazioni supportate si nota come molti software, di norma installati singolarmente su ogni *fat client*, possano essere sostituiti facilmente da un'unica piattaforma web che le renda disponibili dopo un'autenticazione tramite nome utente e parola d'accesso.

E' necessario precisare che la soluzione qui proposta non è in alcun modo collegata al concetto di Terminal Server, che affronteremo in seguito⁸; in quest'ultimo caso, infatti, il server centrale ospita semplicemente delle applicazioni, singolarmente installate sulla macchina, che rende disponibili agli utenti remoti. Nel caso delle web application, invece, la piattaforma è utilizzabile solo tramite server web e browser e non può essere eseguita in un contesto *stand alone* avulso dal network, come nel caso delle applicazioni utilizzate nei *fat client*.

Un approccio di questo tipo alle attività normalmente svolte dalle singole postazioni limita fortemente anche il *vendor tie-in* ed i rischi ad esso

⁸ Cfr. § 4.1.4 dove verrà descritta la tecnologia di Terminal Server.

connessi. L'azienda che facesse largo uso di strumenti di questo tipo, infatti, non si troverebbe dinanzi a particolari problemi in caso di migrazione dei sistemi, cambiamenti nell'infrastruttura di rete o variazioni nella piattaforma utilizzata. Con l'utilizzo di applicazioni web l'unico elemento che garantisce la compatibilità con l'applicazione è costituita dal browser web; nella maggior parte dei casi questi sono assolutamente compatibili tra loro, fatta eccezione per taluni aspetti secondari. Se un software groupware funziona con Internet Explorer, sarà compatibile anche con Mozilla Firefox o con *Safari* per Apple MacOS X. Il grande vantaggio di una configurazione di questo tipo è l'assoluta libertà, indipendenza e flessibilità dei client nei confronti dell'applicazione aziendale utilizzata: qualsiasi sistema dotato di un browser compatibile, ivi compresi i moderni modelli di computer palmari ad esempio, potranno accedere al server centrale dell'azienda e interagire con il groupware.

4. Strumenti e tecnologie per la migrazione

Quanto affrontato nel capitolo precedente è relativo all'analisi ed alla pianificazione da eseguire prima di intraprendere il delicato processo della migrazione dei sistemi.

Nel corso di questo capitolo, invece, illustreremo gli strumenti, le tecnologie e gli approcci da utilizzare per poter *effettivamente* compiere la migrazione alla piattaforma Linux. L'esposizione sarà distinta in sezioni e campi di applicazione, ognuna caratterizzata da peculiarità proprie, da affrontare singolarmente. La sezione che riteniamo essere la più delicata e difficile da pianificare ed implementare è quella relativa alla migrazione delle applicazioni; questo aspetto costituisce nella maggior parte dei casi la barriera principale all'adozione di Linux sui computer client. Verrà affrontata per prima.

4.1 Applicazioni

La migrazione delle applicazioni costituisce nella maggior parte dei casi la barriera più difficile da superare e richiede solitamente un'analisi approfondita e l'elaborazione di strategie di implementazione *ad hoc*. In questo paragrafo affronteremo le più frequenti difficoltà che possono essere incontrate durante il processo di migrazione e suggeriremo tecnologie, strumenti informatici ed alternative per poter raggiungere con successo

l'implementazione di Linux.

Dopo aver condotto l'analisi descritta nel capitolo precedente, è possibile che siano stati identificati delle “strade senza uscita”; talvolta, infatti, possono essere incontrate applicazioni per le quali non esiste un'alternativa sotto il sistema operativo Linux. Un altro scenario potrebbe essere rappresentato da applicazioni alternative che non presentano però le stesse funzionalità delle versioni per Windows. In questa sezione proveremo ad identificare meglio il problema, illustrando alcuni casi e a proporre, quando tecnicamente possibile, una soluzione.

4.1.1 Assenza di funzioni: il caso Microsoft Office

Il primo punto da affrontare è quello dei software Linux ritenuti non “all'altezza” dei corrispondenti per Windows. Nell'ottica di un utilizzo all'interno di un'azienda, riteniamo che il caso di Microsoft Office sia quello principale e che meriti la maggiore attenzione, in questo lavoro e nella strategia di migrazione dei sistemi.

La suite Microsoft è costituita da vari elementi, ognuno dei quali è sostanzialmente diventato lo standard *de facto* nel particolare segmento cui appartiene. Gli applicativi sono rappresentati da *Word* per l'elaborazione dei testi, *Excel* per i fogli di calcolo, *Powerpoint* per le presentazioni, *Access* per i database, *Outlook* per la gestione contatti, posta elettronica e impegni, *Frontpage* per l'elaborazione di documenti HTML. I componenti più utilizzati in ambito aziendale sono senza dubbio Word ed Excel; Powerpoint, rispetto ai primi due, riveste un ruolo secondario, ma

comunque presente. Gli ultimi tre applicativi, Access, Outlook e Frontpage, sono più rari da incontrare nel tipo di postazioni che stiamo esaminando in questo lavoro. Riteniamo, quindi, di dover concentrare la nostra attenzione nella migrazione degli applicativi di elaborazione testi e foglio di calcolo, data anche la notevole semplicità di Powerpoint e le qualità dell'alternativa proposta (*Openoffice Impress*).

Nell'ambito dell'analisi preliminare e della pianificazione di questo processo è necessario valutare, caso per caso, quali siano le funzionalità di questi applicativi utilizzate *effettivamente* dagli utenti finali delle postazioni.

I componenti della suite Microsoft, infatti, sono diventati, con lo sviluppo, sempre più complessi e ricchi di funzioni e possibilità; riteniamo, però, che siano rari i casi di utenti che utilizzino Word o, caso ancora più eclatante, Excel al massimo delle funzionalità offerte o, comunque, in maniera approfondita. E' frequente, infatti, notare che le applicazioni in questione vengano utilizzate per redigere documenti e fogli di calcolo che non fanno in nessun modo ricorso a funzioni particolari o avanzate del software. Per poter quantificare in maniera precisa quale possa essere l'ipotetico danno in termini di produttività del lavoro correlato alla sostituzione di questi prodotti con software alternativi "più semplici", si renderà necessario effettuare preventivamente questa indagine. Così come avvenuto per la classificazione delle applicazioni¹, anche in questo caso la ricerca può essere condotta tramite semplici questionari da presentare agli utenti delle postazioni.

Presumibilmente il risultato dell'indagine mostrerà che la maggior parte di

¹ Cfr. § 3.3 "Ruolo del client e applicazioni utilizzate".

questi farà ricorso alle sole funzioni più elementari e “classiche” di Office e dei suoi applicativi e che non necessiterà quindi di alternative particolarmente complesse, facendo conseguire inoltre un importante risparmio in termini di acquisizione delle licenze². Riteniamo, quindi, che per questo “gruppo di utenti”³, a nostro avviso numericamente superiore in molti scenari, Openoffice costituisca una soluzione assolutamente valida ed implementabile senza difficoltà. La suite in questione, infatti, offre gli stessi componenti del prodotto Microsoft, anche se riteniamo che, soprattutto per quanto riguarda il foglio di calcolo, sia corredato da un minor numero di funzioni e possibilità.

Questo potrebbe causare difficoltà agli utenti più esigenti; il primo riferimento è, a nostro avviso, relativo al *Visual Basic Scripting*, tecnologia proprietaria Microsoft e sulla quale è basato anche Excel, e all'utilizzo complesso delle *macro*⁴ che utilizzano questo linguaggio.

La maggiore difficoltà, in seguito alla migrazione, legata a questa funzione è quella dell'importazione di operazioni macro scritte con Microsoft Office all'interno di Openoffice. Negli ultimi mesi di sviluppo la suite libera ha compiuto importanti passi avanti nel campo della compatibilità dei formati e delle macro, ma ancora non può essere considerata, a nostro avviso,

2 Con dati Febbraio 2005, una singola licenza (per una sola postazione) ha un prezzo che varia dai 198 € della versione *Basic*, comprendente solo Word, Excel e Outlook, a oltre 600 € per la distribuzione *Professional*, completa di tutte le applicazioni. Non sono considerate particolari riduzioni dedicate alle aziende rivenditrici, agli sviluppatori (appartenenti al “*Microsoft Developer Network*”) o specifiche per altri tipi di figure.

3 Notiamo come questi, così come gli “utenti più esigenti” descritti in seguito, possano essere sostanzialmente ricondotti alla classificazione delle postazioni in “Basic Client” e “Intermediate / Advanced Client”, così come descritta nella Tabella 3 del § 3.3 “Ruolo del client ed applicazioni utilizzate”.

4 Operazioni complesse, composte in *Visual Basic*, per l'esecuzione di funzioni e sequenze di operazioni e calcoli; sono utilizzate principalmente, ma non esclusivamente, in Microsoft Excel.

pienamente compatibile con le funzioni più avanzate. Questo potrebbe creare inevitabili problemi, rischiando di minare la produttività di questo particolare gruppo di utenti avanzati, costringendoli da un lato a redigere nuovamente determinati documenti o ad importare le macro “manualmente” e dall'altro a dover rinunciare alle funzioni più complesse. Riteniamo fondamentale notare, però, che le difficoltà in questione non sono relative all'*utilizzo* delle macro, ma esclusivamente all'*importazione* in Openoffice di macro programmate con Microsoft Office. La suite open source, infatti, supporta questo tipo di operazioni, ma utilizza un linguaggio simile a *Starbasic*⁵ che non è compatibile con il Visual Basic utilizzato nei software Microsoft; l'importazione delle macro di Excel in Openoffice, quindi, necessita di una riscrittura del codice stesso della macro. I dati e le funzioni inseriti nel foglio di calcolo, invece, saranno importabili senza difficoltà.

Vi sono aspetti che rendono, però, la suite libera una valida alternativa; tra questi ricordiamo in primo luogo la pressoché totale compatibilità nell'utilizzo dei file scritti con i prodotti Microsoft. Con le dovute precisazioni fatte in merito alle macro e al *Visual Basic Scripting*, non è riscontrabile alcuna difficoltà tecnica nell'importazione dei documenti Microsoft sulla suite open source, rendendo in questo modo possibile l'utilizzo di Openoffice senza correre il rischio dell'impossibilità di scambiare documenti con utenti della suite proprietaria.

Concludendo inoltre l'aspetto legato a Microsoft Office, riteniamo che il costo da sostenere per l'acquisizione delle licenze sia talmente oneroso, considerando anche che la suite in questione è normalmente installata sulla

⁵ Il linguaggio è stato sviluppato da *Sun* per il suo *Staroffice* ed è stato utilizzato in un secondo momento dagli sviluppatori di Openoffice.

quasi totalità delle macchine, da costringere l'amministrazione ad identificare ed adottare soluzioni alternative, per quanto queste possano essere non perfettamente conformi alle aspettative. Considerazioni di questo tipo sono a nostro avviso ulteriormente avvalorate dall'uso che frequentemente viene fatto dei potenti applicativi di Microsoft Office: per banali operazioni di elaborazione testi o di redazione di semplici fogli di calcolo, la suite Microsoft presenta un costo decisamente troppo alto da sostenere, soprattutto nel caso di installazione su numerose postazioni.

4.1.2 Assenza dell'alternativa software per Linux

La maggior parte del software diffuso su larga scala, e qui il riferimento al software commerciale e proprietario è inevitabile, è disponibile quasi esclusivamente per piattaforma Windows o, talvolta, MacOS. Il caso di Microsoft Office è lampante.

Le software house più importanti hanno sempre sviluppato i propri prodotti per il sistema operativo più diffuso e la scelta, quindi, era inevitabilmente Microsoft. Il mercato Apple, invece, ha sempre rappresentato una piccola “nicchia” che ha basato il suo successo nel settore dell'editoria e della grafica, essendo stato, per diversi anni, l'unico sistema dotato delle applicazioni professionali più valide in questo particolare settore. Nel caso di Linux, invece, lo scenario è stato sempre profondamente diverso: la maggior parte del software è stato, ed è, scritto da singoli sviluppatori riuniti dalla collaborazione ad un progetto, reso possibile dalla filosofia open source e dalla licenza di distribuzione GPL. Le *major* del software

difficilmente si sono mai interessate a questa particolare, e poco diffusa, piattaforma, concentrandosi sui sistemi Windows.

Rapidamente lo scenario sta cambiando e importanti attori di questo mercato, dei quali Adobe⁶, Alias⁷, Macromedia⁸, IBM, Novell, Oracle costituiscono solo alcuni esempi, hanno cominciato a sviluppare versioni dei loro software anche per il sistema operativo libero. La situazione, nonostante le ultime evoluzioni, *non è in nessun modo paragonabile* a quella della piattaforma Windows, nettamente più ricca di applicazioni delle più importanti software house e del supporto da esse fornito, ma vi sono importanti premesse per una maggior diffusione⁹.

Nel corso della nostra pianificazione, quindi, sarà necessario identificare le particolari applicazioni per le quali non è prevista nessuna possibile alternativa per la piattaforma Linux; altrettanto fondamentale sarà quantificare il numero delle postazioni che, a causa di ciò, risulterebbero dopo la migrazione ipoteticamente “non operative”. Una volta identificata la gravità del problema è possibile proporre e valutare alcune soluzioni tecniche per rendere comunque possibile, all'interno del processo di migrazione, l'utilizzo di queste applicazioni.

Nel nostro lavoro proponiamo due possibili strategie:

- *Bridging Application*
- Utilizzo dell'applicazione con terminale remoto

6 Celebre per *Photoshop, Pagemaker, Acrobat, Acrobat Reader*.

7 Sviluppatrice di uno dei più importanti software di modellazione 3d, *Maya*.

8 Protagonista nel mercato dei prodotti web e sviluppatrice di *Flash*, oggetto grafico sempre più utilizzato nella progettazione di siti web. I suoi prodotti più importanti sono *Dreamweaver* e *Flash MX*.

9 Affronteremo meglio questo punto nel § 5.4

Le due soluzioni prevedono due approcci profondamente diversi. La prima prevede l'implementazione di Linux sulla postazione client e che l'utilizzo della particolare applicazione avvenga in locale "sotto emulazione"¹⁰. La seconda soluzione, invece, propone l'installazione di una macchina server Windows deputata ad ospitare tutte le applicazioni non disponibili per Linux. Le macchine client, quindi, utilizzando il sistema Linux, accederebbero a queste risorse tramite Intranet *solo* per utilizzare le applicazioni in questione.

4.1.3 Bridging Application

Con questa particolare strategia vengono utilizzati specifici strumenti software in grado di eseguire su Linux applicazioni per Windows. Dal punto di vista tecnico, vi sono principalmente due possibilità: l'emulazione dell'intero sistema e l'emulazione delle *API*¹¹, limitandosi all'esecuzione della sola applicazione.

Le differenze tra le due strategie sono sostanziali, dal punto di vista tecnico e nelle conseguenze pratiche ed economiche. Emulando l'intero sistema, infatti, la macchina è costretta a "ricreare se stessa", emulando quindi la presenza di un *BIOS*¹², di una scheda madre, della memoria e di tutti gli altri

10 Come verrà spiegato successivamente, la principale soluzione tecnica proposta non può essere considerata propriamente una "emulazione" anche se vi assomiglia nelle conseguenze pratiche.

11 *Application Program Interface*, strumento software che si occupa di far comunicare le applicazioni con il sistema operativo.

12 Particolare programma, installato in un chip integrato sulla scheda madre, che viene eseguito all'accensione della macchina e che gestisce, principalmente, la procedura di setup iniziale a livello hardware del sistema.

componenti. Questo approccio, utilizzato ad esempio dal noto *VMWare*¹³, permette l'installazione e l'esecuzione dell'intero sistema operativo da emulare e delle sue applicazioni. Fatta eccezione per il software che si interfaccia direttamente all'hardware, come ad esempio i videogiochi tridimensionali di ultima generazione, richiedendo operazioni e calcoli *direttamente* alle periferiche di sistema, un sistema così emulato permette di utilizzare *qualsiasi* applicazione senza problemi di compatibilità.

Questa importante potenzialità è però fortemente penalizzata dalle richieste di calcolo che software di questo tipo esigono: il carico sulla CPU è notevole proprio perché il sistema è costretto a “ricrearsi” completamente. Le applicazioni utilizzabili, quindi, non sono limitate nel numero da quali siano compatibili, bensì dalle capacità di calcolo del processore centrale del sistema e dalle altre risorse hardware. Le implementazioni pratiche più frequenti per questo tipo di strumento sono quelle di sviluppo applicazioni, *testing* del codice e utilizzo di applicazioni semplici come interfacce per software gestionali e per database server o di applicazioni web.

Una precisazione deve essere fatta in merito ai *costi* di un'implementazione di questo tipo: questo approccio, infatti, presuppone l'installazione del sistema operativo da emulare e, se necessario, delle applicazioni da utilizzare in questo ambiente. I costi di licenza per questi applicativi, quindi, dovranno essere sostenuti *nonostante* la migrazione a sistemi Linux; a questo punto, saranno da applicarsi tutte le considerazioni svolte in precedenza in merito ai costi di licenza¹⁴. Scegliendo questo tipo di strategia si renderà necessario il calcolo dell'incidenza di questi costi sul totale che,

13 Cfr. www.vmware.com

14 Cfr. § 2.4.1 e § 2.4.2.

però, dovrebbe essere minimo dato il presupposto, del nostro lavoro e delle realtà aziendali più comuni, della penuria di applicazioni non implementabili. Con un'emulazione di questo tipo, quindi, i client avrebbero comunque Linux, ma tramite VMWare, o altri software analoghi¹⁵, utilizzerebbero Windows e le applicazioni specifiche per questo sistema operativo.

Qualora determinati utenti debbano fare ricorso a queste con notevole frequenza, siamo dell'opinione che la soluzione più efficiente sia quella di mantenere determinate postazioni con sistema operativo Windows, soprattutto date le limitazioni tecniche insite nell'emulazione di questo tipo, ovvero fare uso di un'altra tecnologia basata, come anticipato precedentemente, sull'emulazione delle *API*. Uno dei vantaggi di questo tipo di implementazione è di non necessitare dell'installazione del sistema operativo "ospite", Windows nel nostro caso. Tramite un particolare software, infatti, risulta possibile l'esecuzione di *determinate* applicazioni Windows all'interno del sistema Linux.

Lo strumento in questione è *Wine*, acronimo per *Wine Is Not An Emulator*, software open source per l'emulazione delle API di Windows; gli sviluppatori hanno riscritto *ex novo* le API del prodotto Microsoft, senza utilizzare nessuna parte di codice sviluppata dall'azienda di Redmond. Questo tipo di approccio nell'emulazione non permette la compatibilità completa con il software Windows, ma limitata a determinati prodotti. La lista delle applicazioni supportate conta attualmente 2525 applicazioni¹⁶, nei

¹⁵ *Bochs*, software open source analogo a VMWare. A differenza di quest'ultimo, *Bochs* non prevede costi di licenza ed è software libero rilasciato con licenza GPL. Cfr. <http://bochs.sourceforge.net/>

¹⁶ Fonte: <http://appdb.winehq.org/> (Febbraio 2005)

più svariati settori: dal software per produttività personale a programmi per il multimedia, dai videogiochi ai software di modellazione tridimensionale. I vantaggi di questo approccio sono diversi e vertono sia sul piano tecnico che su quello economico. La soluzione proposta da Wine, infatti, richiede risorse di calcolo e di hardware nettamente inferiori all'emulazione dell'intero sistema, così come con Bochs o VMWare, ed assolutamente paragonabili a quanto richiesto dal sistema operativo "nativo", Windows. Il software, infatti, non deve "ricostruire" l'intero sistema e potrà fare uso delle risorse del computer e delle sue periferiche, anche se non supportate dal "sistema operativo emulato". Le applicazioni Windows utilizzate sotto Wine fanno ricorso al sistema operativo, Linux nel nostro caso, per l'interfacciamento con hardware e periferiche di sistema, permettendo un notevole alleggerimento del lavoro richiesto alla CPU e alla memoria e garantendo prestazioni analoghe a quelle conseguite con Windows.

Dal punto di visto tecnico, quindi, Wine è più conveniente dell'emulazione "dell'intero sistema"; l'approccio in questione, però, limita la compatibilità delle applicazioni ad un numero inferiore rispetto a quanto permesso da VMWare o Bochs. In quest'ultimo caso, come visto, la compatibilità è pressoché completa e sono esclusi solo quei software che facciano diretto uso di risorse hardware (come ad esempio le applicazioni 3D); nel caso di Wine, invece, *non è prevedibile a priori* la compatibilità di una determinata applicazione e deve essere verificata caso per caso.

Ai fini dell'implementazione di questo strumento in azienda per permettere l'utilizzo delle applicazioni Windows in ambiente Linux, si renderà quindi necessaria una documentazione ed un'analisi della resa di questi software

utilizzati con Wine.

Per gli intenti del nostro lavoro riteniamo che, tra le applicazioni compatibili con Wine, le più importanti siano rappresentate da *Microsoft Office*, *Macromedia Dreamweaver*, *Microsoft Project*, *Adobe Photoshop*, *VisualFox Pro*, *Microsoft Internet Explorer*, *Microsoft Outlook*. Concentrandoci sulla suite Microsoft, possiamo affermare che l'utilizzo tramite Wine delle applicazioni rende pressoché inavvertibile all'utente l'esecuzione “sotto emulazione”, garantendo quindi la produttività delle postazioni e la non interruzione delle attività.

Questa strategia di implementazione pone comunque il problema del sostenimento delle licenze per le singole applicazioni per Windows utilizzate, permettendo solamente di evitare i costi legati alla licenza del sistema operativo Microsoft. Siamo, quindi, dell'opinione che una strategia di questo tipo possa risolvere le eventuali difficoltà conseguenti l'incompatibilità di determinati software necessari per lo svolgimento dell'attività, ma riteniamo in ogni modo che si debba provvedere, ove possibile, con altri strumenti che illustriamo nel nostro lavoro.

Dall'esperienza degli sviluppatori di Wine, poi, è nato il progetto di *Codeweavers*, azienda statunitense specializzata nel rendere utilizzabili applicazioni Windows sotto Linux, con un focus principale mirato alle esigenze del mondo aziendale. Utilizzando il codice sorgente di Wine, rilasciato con licenza GPL, hanno sviluppato il proprio prodotto concentrandosi sulle applicazioni Office e di produttività. Il software, *Crossover Office*, migliora la compatibilità di determinate applicazioni e garantisce, inoltre, supporto prestato direttamente da Codeweavers, a fronte

di un costo di licenza limitato.

Una tecnologia interessante sviluppata da questa azienda è quella utilizzata nel loro prodotto dedicato al mondo server e alla multiutenza in *thin client*. Il fine è quello di accentrare l'esecuzione delle applicazioni Windows su di un server centrale con Linux e Crossover Office. In questo modo le singole postazioni non necessitano di alcun software di emulazione, ma solamente di un collegamento di rete che li connetta al server centrale: sarà quest'ultimo a rendere disponibili le singole applicazioni ai vari utenti. Nel paragrafo successivo proponiamo appunto una soluzione basata su un approccio di questo tipo, ancorché indipendente dal software di *Codeweavers*.

4.1.4 Utilizzo dell'applicazione tramite Terminal Server

Un'alternativa alle strategie “di emulazione” sopra descritte è costituita dall'esecuzione in remoto delle applicazioni, analogamente al caso di *Crossover Office Server* trattato al termine del paragrafo precedente. Ciò che vogliamo illustrare in questo paragrafo, però, differisce dal prodotto derivato da Wine per un aspetto fondamentale: il server centrale che rende disponibili le applicazioni ai computer client non sarà un sistema Linux, ma un sistema Windows.

Un'implementazione di questo tipo può rivelarsi necessaria e particolarmente utile in più occasioni:

- incompatibilità completa dell'applicazione Windows con

Linux, anche in emulazione API con Wine

- impossibilità di eseguirla con software come VMWare a causa delle richieste di calcolo o altre motivazioni
- importanza dell'applicazione per il normale svolgimento delle attività, tale da rendere preferibile un'implementazione nativa

Riteniamo che l'accesso in remoto delle applicazioni sia una soluzione valida in configurazioni che presentino queste caratteristiche; in tal modo, infatti, l'implementazione del sistema Linux potrebbe avvenire sulla totalità delle macchine client e le applicazioni non compatibili sarebbero installate su una *singola* macchina, il server appunto, che le rende disponibili tramite rete.

L'accesso dei client al server può avvenire tramite diversi strumenti: da *Windows Terminal Server*, tecnologia proprietaria di Microsoft, a *VNC*, open source e ideale per applicazioni non pesanti (ad esempio, interfacce di accesso a database) o il nuovo e performante *NX Terminal Server*, sviluppato dall'azienda italiana *NoMachine* e distribuito, in parte, con licenza GPL. Il meccanismo di funzionamento con un simile approccio è molto semplice e allo stesso tempo potente: l'utente di una postazione client dovrà avviare il programma di collegamento al server, autenticarsi con nome utente e parola d'accesso e procedere con le operazioni come se fosse davanti alla macchina Windows. Una volta concluso il lavoro con la specifica applicazione, è sufficiente terminare il collegamento al server e continuare il lavoro con la propria postazione Linux. La velocità e

l'affidabilità raggiunta da questi prodotti, soprattutto *NX Terminal Server*, è tale da non percepire, nel normale svolgimento delle operazioni, l'utilizzo di una macchina remota invece che del proprio personal computer, a fronte di un modesto consumo di banda per il collegamento.

In relazione a questa strategia di implementazione è necessario precisare quali siano i termini di licenza dei software resi disponibili tramite rete e *Terminal Server*; le condizioni in questione possono variare sensibilmente da produttore a produttore¹⁷. In linea generale è possibile delineare un principio comune che prevede il pagamento della licenza per ogni postazione che possa *potenzialmente* eseguire l'applicazione. Non è in nessun modo contemplata la possibilità di distinguere il momento dell'esecuzione da postazione a postazione, rendendo in questo modo inapplicabile un pagamento in base all'utilizzo *nello stesso momento* dell'applicazione. E' di norma consentito lo spostamento della licenza da una macchina ad un'altra, qualora sia previsto che la prima non debba più far uso del software in questione, ma non è possibile farlo in base ad esigenze momentanee; questo implica che qualora il software sia utilizzato saltuariamente e, magari, da non più di due utenti alla volta, sarà necessario pagare la licenza per ogni singola postazione che eseguirà l'applicazione.

In una configurazione che preveda un accesso via rete alle risorse di un server centrale il concetto di “eseguire un'applicazione” contempla naturalmente anche il caso costituito dall'accesso in remoto all'applicazione stessa; è fondamentale, quindi, valutare quante postazioni faranno uso del software installato sul server centrale per poter determinare i costi di licenza

¹⁷ A riguardo, sarà necessaria la consultazione delle *EULA, End User Licence Agreement* del software.

legati ad una strategia di questo tipo.

L'esempio su descritto delle condizioni di licenza è stato analizzato in base alle EULA dei software Microsoft, che però rispettano linee generali seguite anche da altri produttori; è comunque necessaria un'analisi specifica per le singole applicazioni per poter meglio valutare l'impatto di una simile strategia. Naturalmente, il mantenimento di Windows su determinate postazioni client comporterebbe comunque un costo di licenza superiore all'implementazione tramite Terminal Server; la differenza tra le due soluzioni avrà, come limite inferiore, il costo delle singole licenze per il sistema operativo delle postazioni Windows.

Quanto in precedenza illustrato in riferimento alle applicazioni non implementabili sotto Linux, ha mostrato che qualsiasi strategia utilizzata per rendere disponibili questi software mitiga di fatto l'effetto del contenimento dei costi di licenza realizzato con la migrazione. Sia il *bridging application* che l'accesso ad un server centrale tramite *Terminal Server* presenta l'esigenza di acquistare le licenze per le applicazioni commerciali utilizzate; è stato inoltre illustrato come, di norma, le condizioni di utilizzo prevedano l'acquisto per ogni singola postazione debba far uso del software. A nostro avviso una configurazione che faccia uso di queste strategie è comunque sostenibile perché permette allo stesso tempo l'utilizzo delle applicazioni critiche e il conseguimento dei vantaggi derivanti dall'implementazione di Linux in termini di sicurezza, affidabilità, indipendenza e flessibilità.

4.1.5 Le alternative software

Come descritto nel terzo capitolo, durante l'analisi preliminare è opportuno indagare su quali siano le applicazioni più utilizzate dalle diverse postazioni e quale sia l'uso che ne venga fatto, distinguendo quindi in diversi livelli di approfondimento. Ad un livello più avanzato corrisponderanno naturalmente maggiori esigenze da soddisfare dalla futura alternativa software per Linux e si dovrà quindi prestare più attenzione nelle scelte da compiere e nella configurazione dei sistemi.

Una volta identificati i “software critici” si ritiene opportuno redigere un prospetto nel quale proporre l'applicazione alternativa da utilizzare e le considerazioni relative ad eventuali problemi, così come indicato in Tabella 4. Nella rappresentazione proposta di seguito viene anche indicata l'eventuale presenza dello stesso software in versione nativa per Linuxe la possibilità di utilizzare la versione per Windows.

<i>Applicazione originaria</i>	<i>Versione nativa per Linux ?</i>	<i>Alternativa post migrazione</i>	<i>Problemi</i>
Interfaccia Utente e Grafica di Windows	NO	Gnome KDE XFCE	Aspetto diverso Menù contestuali Diversi path dei file
Internet Explorer	NO*	Mozilla Firefox KDE Konqueror Mozilla Navigator Opera Browser	Aspetto diverso
File Explorer	NO	KDE File Manager Gnome Nautilus Rox-filer XF File manager	Menù contestuali Diversi path dei file
Microsoft Outlook	NO*	Mozilla Sunbird Gnome Evolution	Aspetto diverso
Microsoft Outlook Express	NO*	Mozilla Thunderbird KDE Mail Sylpheed-claws Gnome Evolution	Aspetto diverso
MSN Messenger	NO	Gaim Jabber	-
Skype, VoIP software	SI'	Skype for Linux	
Windows Media Player	NO	Xine Mplayer Totem	-
Microsoft Office	NO*	Openoffice.org Koffice	Aspetto diverso Funzionalità limitate nell'importazione delle macro
Riproduttore musicale (Winamp)	NO*	Xmms Xine Totem	-
Archiviatore (Winzip, Winace, Winrar)	SI'	Ark File Archiever File-roller	-

<i>Applicazione originaria</i>	<i>Versione nativa per Linux ?</i>	<i>Alternativa post migrazione</i>	<i>Problemi</i>
		zip / unzip ace / unace rar / unrar	
Visualizzatore immagini (ACDSee)	NO*	GQView	-
Fotoritocco e grafica (Adobe Photoshop)	NO*	the Gimp	-
Adobe Acrobat Reader	SI'	Adobe Reader for Linux	-
Editor HTML (Macromedia Dreamweaver, Frontpage)	NO*	Nvu Macromedia Dreamweaver* Mozilla Composer	-

Tabella 4: Applicazioni originarie, le possibili alternative sotto Linux e considerazioni relative

** possibile implementazione con emulatore API per Linux (Wine)*

Nella Tabella 4 abbiamo inserito le applicazioni che riteniamo essere più frequentemente utilizzate in ambito aziendale; per ognuna di esse esistono diverse alternative rappresentate da software per Linux. In alcuni casi sono state anche inserite applicazioni molto diffuse e che possono rappresentare il caso di realtà aziendali operanti in uno specifico settore (fotoritocco, grafica editoriale, editor HTML).

Notiamo nell'ultima colonna relativa ai "problemi di migrazione" che, nella maggior parte dei casi, le alternative per Linux sono caratterizzate da un aspetto diverso rispetto alle applicazioni originarie per Windows, talvolta da un'assenza di alcune funzionalità o da aspetti marginali (menù contestuale o percorso dei file e delle cartelle più comuni). In alcuni casi, ad esempio per Microsoft Office o Internet Explorer, le applicazioni originarie per

Windows sono anche utilizzabili, tramite particolari strumenti, in un sistema Linux: questo punto è stato affrontato nel corso dei paragrafi precedenti relativi all'implementazione di applicazioni per Windows sotto Linux, illustrando le possibilità offerte dal Bridging Application o dall'accesso tramite Terminal Server.

Tramite un'analisi di questo tipo sarà possibile avere dinanzi una rappresentazione chiara dello scenario aziendale delle postazioni informatiche e poter realizzare una pianificazione per la migrazione dei sistemi. A tal fine si rende necessario identificare quali debbano essere le contromisure da adottare per superare, o mitigare, i problemi delle applicazioni alternative. Come notiamo nella Tabella 4, non vi sono problemi particolari nell'utilizzo dei software indicati per Linux, se non nell'aspetto delle applicazioni che può differire da quella originaria per Windows. Qualora la differenza sia sensibile e sia sostanzialmente limitata all'interfaccia utente, sarà necessario prevedere una documentazione, possibilmente resa disponibile tramite Intranet a tutte le postazioni, che illustri le divergenze più importanti e che spieghi il comportamento dei nuovi software. Ad un particolare aspetto di questo problema, relativo all'interfaccia utente del sistema operativo, dedicheremo il paragrafo seguente, illustrando anche quali possano essere gli strumenti per rendere la migrazione meno problematica per i nuovi utenti.

4.2 Interfaccia utente

L'interfaccia utente è lo strumento con il quale l'utente entra in contatto,

interagisce ed utilizza il personal computer. I sistemi operativi solitamente presentano una propria interfaccia utente che li distingue dagli altri prodotti e che è caratterizzata da un comportamento leggermente differente. Apple, ad esempio, usa un sistema operativo, MacOS, che differisce da Windows o da Linux per molti aspetti: dall'interfaccia per la gestione dei file all'interazione tramite mouse, dal desktop alle impostazioni di sistema. Dopo un utilizzo, magari quotidiano, di un determinato sistema operativo, è normale che l'utente si abitui a determinate operazioni e comportamenti caratteristici di quel sistema: con un esempio banale, un utente Windows cercherà sempre le applicazioni partendo dal pulsante “*Start*” in basso a sinistra del proprio schermo. Se dinanzi ad un sistema Apple, egli dovrà intuire, o cercare, quale sia la tecnica caratteristica con la quale accedere alle applicazioni in MacOS.

Linux differisce da Windows o MacOS anche sotto questo aspetto: le interfacce disponibili per Linux, o meglio per il suo server grafico¹⁸, sono decine, spesso con caratteristiche molto diverse le une dalle altre.

Ciò che per qualche utente può creare disorientamento, diventa però un potente strumento nell'affinare la migrazione verso Linux. Un tale livello di personalizzazione, infatti, rende possibile all'amministrazione di modificare a piacimento il comportamento dell'interfaccia in tutte le postazioni,

18 Lo strumento per disporre sotto Linux di una interfaccia grafica, basata su finestre ed utilizzo del mouse, è il cosiddetto “server grafico”: software basato su un'architettura client/server, è ciò che rende possibile al sistema di usare un dispositivo grafico, diverso dalla console di testo. Storicamente, l'applicazione in questione è stata *Xfree86 Server*; di recente, a causa di problemi nel cambiamento di licenza seguito dagli sviluppatori, è stato sostituito da molte distribuzioni Linux da *X.org*. Ciò che poi differenzia l'aspetto dell'interfaccia grafica non è il server grafico (comune per tutte) ma il “gestore di finestre” (*window manager*) o “ambiente desktop” (*desktop environment*): questi software sono costituiti da un insieme di applicazioni e strumenti in grado di gestire le finestre, il desktop, le impostazioni di mouse e tastiera, la navigazione tra file e directory e altri aspetti propri dell'interfaccia.

configurandone tutti gli aspetti. Esistono, ad esempio, "gestori di finestre" (*window manager*) che copiano fedelmente il comportamento e l'aspetto di Windows, dalle versioni più vecchie al recente XP, o di MacOS o, ancora, che utilizzano un'interfaccia tridimensionale mai utilizzata da nessun altro sistema operativo¹⁹. Gli "ambienti desktop" (*desktop environment*), invece, sono degli strumenti più complessi ed articolati e che non si limitano alla sola "gestione delle finestre"; prevedono spesso applicazioni per la configurazione del sistema, gestione delle applicazioni, software aggiuntivi per il desktop, applicazioni per la navigazione dei filesystem, browser web e altri strumenti. Tra questi i più famosi sono senza dubbio *Gnome* e *KDE*²⁰, progetti che hanno raggiunto la piena maturità e che attualmente offrono funzionalità analoghe, e in taluni aspetti superiori, alle interfacce grafiche di Windows e MacOS.

Ciò che abbiamo premesso è però assolutamente fondamentale: ad un semplice utente spesso non interessa che il nuovo strumento sia migliore o più potente se non è in grado di utilizzarlo e sfruttarlo al meglio. Nel corso del processo di migrazione diventa quindi necessaria un'attenta pianificazione e una progettazione dell'interfaccia al fine di rendere il passaggio il più possibile trasparente e assolutamente non traumatico. Il nostro lavoro illustrerà successivamente alcuni strumenti e tecniche al fine di conseguire questo obiettivo garantendo, in questo modo, la non interruzione o rallentamento delle attività svolte dalle postazioni.

¹⁹ Quest'ultimo caso è frutto di un progetto di Sun, *Looking Glass*, attualmente rilasciato ancora come versione di sviluppo e non considerata ancora stabile. Abbiamo riportato questo esempio per sottolineare quale sia la varietà possibile dei gestori di finestre in ambiente Linux e come, quindi, possa essere modificata profondamente l'interfaccia grafica dello stesso sistema. Cfr. www.sun.com/software/looking_glass/demo.html

²⁰ Cfr. www.gnome.org e www.kde.org

Affronteremo due aspetti che, se non adeguatamente considerati, possono creare difficoltà agli utenti e rendere la migrazione suscettibile di insuccesso a causa delle resistenze di questi ultimi e degli effettivi problemi incontrati. La nostra esposizione non farà diretto riferimento agli strumenti software da utilizzare, in quanto la maggior parte di quanto qui proposto è applicabile a diversi “gestori di finestre” o "ambienti desktop"; l'amministrazione potrà quindi decidere quale debba essere il software più adatto ed implementarlo secondo le proprie esigenze. A nostro avviso, però, Gnome e KDE costituiscono le due alternative più importanti, in quanto a maturità, rapidità di sviluppo e dimensione della comunità di sviluppatori.

4.2.1 Interfaccia grafica e aspetto del desktop

L'interfaccia grafica del sistema operativo costituisce senza dubbio un punto molto delicato nella migrazione a piattaforma Linux. A nostro avviso è sconsigliabile l'utilizzo di interfacce grafiche particolarmente innovative o complesse; per quanto ricche di funzionalità, potrebbero disorientare l'utente e conseguire quindi il risultato opposto rispetto a quanto prefissato. Le impostazioni del *desktop environment* dovrebbero seguire le direttive dell'ambiente Windows, tentando di avvicinarsi quanto più possibile al funzionamento del sistema Microsoft. I permessi necessari alla personalizzazione dell'interfaccia dovrebbero essere negati all'utente comune nella prima fase della migrazione, al fine di permettere agli utenti

di concentrarsi su una sola interfaccia, scelta dall'amministrazione. Questa limitazione è a nostro avviso importante anche nell'ottica del supporto tecnico che dovrà essere fornito: qualora tutte le postazioni presentino le stesse caratteristiche di funzionamento generale, sarebbe più semplice e meno dispersivo prestare assistenza a chi si trovasse in difficoltà. Solo successivamente, quando sarà acquisita la necessaria padronanza del nuovo sistema, potrà essere concesso agli utenti comuni di personalizzare come meglio preferiscono il funzionamento dell'interfaccia.

Particolare attenzione deve essere rivolta al comportamento e alle impostazioni del desktop: questo deve essere il più possibile simile a quanto incontrato sui sistemi Windows. Gli strumenti accessibili dovranno quindi essere i medesimi: esplorazione del filesystem (*“Risorse del computer”* sui sistemi Microsoft), esplorazione delle risorse condivise via rete locale (*“Risorse di rete”*), *“Cestino”*, browser web, client di posta elettronica e cartella personale dei documenti (*“Documenti”*). La possibilità di inserire collegamenti ad applicazioni, file o directory su desktop deve seguire un funzionamento analogo a quello di Windows: la pressione del tasto destro del mouse dovrà permettere l'operazione, insieme ad altre funzioni analoghe. Allo stesso modo dovrà essere permesso all'utente di personalizzare l'aspetto grafico del desktop, qui inteso come immagine di sfondo, colori e caratteri. L'accesso alle applicazioni installate avviene, come noto, tramite un pulsante in basso a sinistra dello schermo; sarà così

preferibile scegliere un'impostazione del desktop tale da permettere un funzionamento analogo ed una disposizione delle funzioni accessibili simile a quanto presente in Windows; l'utente, quindi, potrà accedere con semplicità alla configurazione dell'aspetto grafico, alla ricerca di file e directory, alle applicazioni installate, ai dati utilizzati recentemente e così via.

Un'altra caratteristica di Linux che dovrebbe essere celata agli utenti nella prima fase della migrazione è quella della ricchezza di applicazioni disponibili per una determinata funzione; talvolta, infatti, utenti abituati alla presenza di una sola applicazione “per categoria” potrebbero essere disorientati qualora avessero una possibilità di scelta troppo vasta. Nella fase preliminare alla migrazione, durante l'analisi dello scenario e la pianificazione dei software alternativi da utilizzare in Linux, si dovrà quindi procedere ad un accurato esame delle possibilità e procedere con la scelta di un solo software dedicato ad una particolare funzione. In questo modo tutte le postazioni, o categoria di queste, utilizzeranno gli stessi software e il supporto tecnico potrà concentrarsi su essi, redigendo, ad esempio, della documentazione dedicata. La possibilità di installare nuove applicazioni, o scegliere tra diverse alternative, dovrebbe essere quindi concessa solo in un secondo momento, analogamente a quanto visto per la personalizzazione dell'interfaccia.

4.2.2 Differenze terminologiche e percorsi dei file

I sistemi operativi, frutto dello sviluppo da parte di persone, aziende e contesti diversi, hanno, con il passare del tempo e con l'affermazione sul mercato, creato un “vocabolario” proprio che li distingue gli uni dagli altri. Talvolta, quindi, può capitare che un utente si trovi dinanzi a qualcosa che conosce, ma che viene però denominata in maniera diversa, creando confusione e disorientamento. Situazioni analoghe possono presentarsi anche a soggetti abituati ad utilizzare software in inglese, la “lingua madre” dell'informatica, che si trovino dinanzi alle traduzioni in italiano dell'interfaccia: spesso termini conosciuti universalmente nella loro versione inglese vengono tradotti in maniera non accurata, o talvolta assolutamente sbagliata, confondendo anche utenti esperti.

Nei sistemi Windows un esempio lampante è costituito dal termine “*cartella*” (*folder* in inglese): in campo informatico, l'espressione in questione non esiste. File e *directory* sono oggetti gestiti da tutti i sistemi operativi, anche precedentemente alla diffusione del sistema Microsoft; per quest'ultimo, però, fu scelto il neologismo “*cartella*”, indubbiamente più intuitivo e più familiare nell'approccio, ma che si scosta sensibilmente dallo standard. Parallelamente alle vendite e alla diffusione di Windows, questa espressione divenne un sinonimo ed in tempi recenti, per molti utenti, l'unico termine utilizzabile per definire il concetto di *directory*.

La configurazione delle diverse postazioni deve tener conto delle differenze terminologiche, corrette o meno che siano, esistenti tra Windows e Linux, ricordando sempre che l'obiettivo principale rimane quello di permettere una migrazione “dolce” senza creare inutili difficoltà. Dal punto di vista tecnico questo fine si riflette sulla scelta di distribuzioni Linux che presentino la traduzione in italiano del sistema, dell'interfaccia grafica e delle applicazioni specifiche.

L'aspetto della localizzazione in italiano del software può non essere sufficiente: particolare importanza deve essere dedicata a tutte le risorse, gli oggetti, gli strumenti utilizzati frequentemente in Windows e che abbiano in Linux nomi diversi. La prima differenza sostanziale è incontrata con la denominazione dei dispositivi e filesystem: nei sistemi Microsoft questi assumono come nome le lettere dell'alfabeto. Abbiamo così il lettore di dischetti “A:”, la prima (e spesso unica) partizione del disco fisso “C:”²¹, il lettore CD o DVD “D:” e proseguendo allo stesso modo con gli altri dispositivi. In Linux, invece, l'assegnazione del nome è profondamente diversa e più difficile da memorizzare per l'utente, anche se più precisa: la gerarchizzazione, infatti, è assegnata in base all'hardware e alla connessione dei dispositivi²².

21 Storicamente la lettera “B:” era destinata al secondo lettore di dischetti; in tempi recenti questo tipo di supporto è diventato obsoleto a causa delle limitazioni tecniche, ma ciò nonostante è stata conservata la tradizione di riservare le prime due lettere dei dispositivi ai *floppy*.

22 Più precisamente, i sistemi Unix assegnano nomi di unità in base ai canali di connessione utilizzati: per le periferiche *IDE*, le più diffuse, la distinzione avviene in base ai canali “primario” e “secondario” e, per ognuno di essi, “*master*” e “*slave*”. Così facendo, un disco fisso collegato al canale primario, in posizione master, sarà identificato come “*hda*”, il secondo sullo stesso canale “*hdb*” e così via con “*hdc*” e “*hdd*” per il canale secondario. Le periferiche *SCSI*, invece, utilizzano nomi di device del tipo “*sd*” o “*sg*”, seguite da lettere o

Questo crea naturalmente confusione per l'utente Windows e, spesso, difficoltà nell'identificazione delle diverse periferiche connesse al sistema; la soluzione a questo tipo di problema è molto semplice e non ha bisogno di particolari accorgimenti tecnici. Possono essere creati semplici *link simbolici* ai dispositivi (una sorta di *alias* con i quali il sistema li identifica, utilizzando le denominazioni tipiche del sistema Windows; così facendo, l'utente potrà accedere ad “hda1” semplicemente cliccando su un link dal nome “C:” e analogamente con il lettore cd o floppy e con tutti gli altri dispositivi collegati alla postazione.

Quanto finora descritto è relativo ai soli dispositivi di memorizzazione; un'altra importante differenza tra Windows e Linux è invece legata ai percorsi (“*path*”) di file e directory usati più frequentemente. L'esempio più comune è rappresentato dalla cartella “Documenti” di Windows, la porzione del filesystem (normalmente, “C:”), dove vengono memorizzati la maggior parte dei dati e dei file personali dell'utente. Questi, insieme alle impostazioni e le configurazioni per ogni account, sono memorizzate in una cartella con un percorso particolare²³. Nei sistemi Linux, invece, la disposizione dei file è diversa e, in questo caso, più razionale: la directory relativa ad ogni utente è contenuta nella sua directory “*/home/nome_utente/*”, chiamata appunto “*home directory*”. Al suo interno sono memorizzati *tutti* i file propri dell'utente e relativi alle sue

numeri identificativi.

²³ Generalmente, nei sistemi XP, *C:\Documents and Settings\nome_utente*.

configurazioni ed impostazioni; le applicazioni, infatti, dopo la prima esecuzione, creano normalmente una directory nascosta all'interno della “home directory” dell'utente, dove memorizzeranno le configurazioni specifiche. Su questa parte del filesystem l'utente proprietario ha il pieno controllo ed è l'unica “sezione” della quale può disporre liberamente, salvo particolari configurazioni di sistema: i vantaggi di una tale impostazione in termini di sicurezza sono notevoli, permettendo una configurazione del sistema in multiutenza “a compartimenti stagni”²⁴.

I documenti personali di ogni utente potrebbero essere contenuti nella sua stessa directory, o in una specifica cartella al suo interno; a riguardo, l'amministrazione dovrà anche in questo caso considerare la differenza terminologica, creando, ad esempio, un link simbolico *ad hoc* “Documenti”. Un approccio del genere dovrà, a nostro avviso, essere seguito anche nella configurazione degli altri strumenti e risorse più frequentemente utilizzate dagli utenti: ad esempio, per la navigazione di file e cartelle all'interno del proprio computer (“Risorse del computer”) o per l'esplorazione delle risorse condivise tramite rete (“Risorse di rete”²⁵).

4.3 Server e infrastruttura di rete

La migrazione a Linux dei sistemi client costituisce un passo importante per

²⁴ Abbiamo approfondito l'argomento nel corso del secondo capitolo. Cfr. § 2.1.3.

²⁵ Cfr. § 4.3.1 e § 4.3.2.

il conseguimento dei vantaggi offerti in termini di costi di licenza, flessibilità, facilità di manutenzione e *vendor tie-in*, affrontati nel corso del secondo capitolo. Abbiamo già illustrato come la fase più delicata e complessa di un processo di questo tipo sia costituita dalla migrazione dei sistemi desktop, delle postazioni dalle quali i singoli utenti dovranno svolgere il proprio lavoro: per queste ragioni la parte principale del nostro lavoro si concentra su questo aspetto.

I sistemi informatici convivono sempre più spesso in un contesto di *rete*, network con architettura server client che rendono disponibili alle singole postazioni determinate risorse: da applicazioni particolari a filesystem condivisi, da stampanti di rete ad interfacce per il collegamento con server database per l'utilizzo di determinate applicazioni, come ad esempio nel caso dei software gestionali.

Nel corso di questo paragrafo ci occuperemo brevemente di quali possano essere gli strumenti e le tecnologie da adottare al fine di implementare Linux anche sulle macchine server di un network aziendale. Alcune di queste si riveleranno necessarie per lo svolgimento di determinate operazioni, come ad esempio la condivisione di filesystem di rete²⁶, in un ambiente caratterizzato dalla presenza di macchine Windows e Linux.

Il sistema operativo libero ha conosciuto, fino ai tempi più recenti, la sua maggior diffusione e successo principalmente nell'ambito di macchine server, destinate allo svolgimento di un determinato servizio. L'esempio di

26 Cfr. 4.3.1.

Apache web server, affrontato nel corso del secondo capitolo²⁷, costituisce uno dei casi più importanti di software open source di qualità per macchine server e che abbia riscosso notevole successo. Alla luce della notevole presenza di applicazioni server per Linux, riteniamo che la migrazione anche di questi sistemi possa costituire un importante passo e possa far conseguire importanti vantaggi, seppur minori di quelli legati alla migrazione dei sistemi client. Le motivazioni alla base sono principalmente legate alla proporzione di macchine server sul totale in un network e agli utenti che faranno uso di questi sistemi.

Il primo punto è strettamente legato al concetto stesso di network: in contesti di questo tipo singoli server prestano servizi e rendono disponibili risorse ad una moltitudine di client che vi accedono. I vantaggi in termini di risparmio sui costi di licenza, nel caso di migrazione a Linux dei soli server, saranno quindi limitati al ristretto gruppo di macchine server all'interno del network; l'istituzione continuerà ad avere importanti inefficienze legate alle singole postazioni, ovvero la maggior parte del proprio parco macchine.

Particolarmente importante è anche l'effetto che una migrazione parziale, destinata alle sole macchine server, potrebbe avere sulla manutenzione dei sistemi. Come affrontato nel secondo capitolo, l'implementazione di Linux sulle postazioni client può permettere una notevole riduzione delle risorse necessarie alla configurazione dei sistemi, alla loro amministrazione e manutenzione; migrando i soli sistemi server, però, non sarà possibile

²⁷ Cfr. 2.4.2.

conseguire detti vantaggi per la maggior parte del network.

Il secondo aspetto, legato alla tipologia di utenza, è strettamente correlato alle tematiche di sicurezza affrontate nel secondo capitolo²⁸. Abbiamo, infatti, illustrato come le maggiori diffusioni di “malware” sono state frutto di attacchi diretti a macchine desktop e non a server di produzione. Fatta eccezione per “*SQL Slammer*”²⁹ ed altri di minor rilievo, i worm più dannosi sono stati scritti per colpire i sistemi degli utenti comuni, normalmente meno difesi e configurati in maniera più superficiale, al fine di generare e diffondere l'infezione. I server, sia che utilizzino Linux o Windows, sono, o quanto meno dovrebbero essere, configurati con il focus principale rivolto alla sicurezza e alla stabilità del servizio; lo scenario dei sistemi desktop è invece caratterizzato dall'*assenza completa* di protezioni software o particolari accorgimenti tecnici. Una migrazione parziale che escludesse le postazioni client presenterebbe, quindi, uno squilibrio nella sicurezza e nell'efficienza del network. La diffusione di “malware” all'interno di una rete locale, infatti, anche se non in grado di colpire altri sistemi, può creare notevoli rallentamenti alla rete e rendere la stessa inutilizzabile a causa del traffico generato e delle frequenti connessioni effettuate.

L'intento del nostro lavoro è quello di illustrare al meglio il processo di migrazione dei sistemi client, alla luce degli importanti effetti che questo

28 Cfr. § 2.1.

29 Cfr. § 2.1.

può avere sulla gestione dei sistemi e dell'intero network aziendale; la migrazione dei sistemi server, invece, presenta peculiarità e caratteristiche di implementazione tali da far differire notevolmente i diversi scenari di applicazione. Per completezza, però, affronteremo alcuni strumenti software necessari, durante e dopo la migrazione, all'integrazione dei client all'interno del network, distinguendoli in base alla funzione.

4.3.1 Convidisione in rete di file e directory

L'utilizzo di tecnologie server client presuppone l'esistenza di determinate risorse condivise, siano esse rappresentate da filesystem di rete, stampanti o portali web, rese disponibili da singoli sistemi che prestano servizi (server) e utilizzati da una pluralità di macchine (client). La configurazione più comune di un network presenta così un accentramento nello svolgimento di funzioni, come nel caso della memorizzazione dei dati in un database installato su una macchina server o nella condivisione di file e directory. Quest'ultima è una caratteristica sempre più diffusa nell'ambito di network aziendali, al fine di rendere possibile una collaborazione nella stesura di documenti da parte di più soggetti o il semplice accesso ad essi da parte di più postazioni.

Nell'ambito dei sistemi Linux il software utilizzato più frequentemente è “*NFS (Network File System) Server*”. Sviluppato originariamente nel 1984,

permette la condivisione di file e directory accessibili da numerose macchine come se fossero memorizzati localmente. Il progetto è storicamente legato ai sistemi Unix e *Unix like*, ma esistono implementazioni anche per Windows o Apple MacOS. Il protocollo analogo, sviluppato da Microsoft per i propri sistemi, è *SMB, Server Message Block*.

La condivisione di risorse all'interno di una rete mista Windows Linux, però, deve essere effettuata tramite un altro software open source che ha riscontrato notevole successo per la qualità del codice, la velocità e le funzionalità offerte: “*Samba*” costituisce l'implementazione open source del server SMB di Microsoft³⁰: tramite questo strumento, infatti, le macchine Windows all'interno di una rete locale possono condividere file e cartelle con sistemi Linux, senza accorgersi, nell'utilizzo, di alcuna differenza rispetto a quanto offerto dal server Microsoft. Samba, inoltre, è particolarmente apprezzato all'interno di importanti network per le sue doti di scalabilità che rendono possibile la condivisione tra centinaia di macchine senza i problemi di lentezza e stabilità delle connessioni tipiche di altri prodotti. Questo strumento permette anche l'applicazione sulle risorse condivise di politiche di gestione dei permessi che si aggiungono a quanto determinato tramite filesystem locale; in questo modo, è possibile distinguere i permessi accordati agli utenti locali rispetto a quanto

³⁰ Lo stesso nome *Samba* è originato, a detta degli autori, dalla semplice aggiunta di due vocali al nome del software Microsoft.

configurato per gli utenti di rete, permettendo in tal modo una configurazione più flessibile.

4.3.2 Condivisione di stampanti

L'utilizzo di stampanti di rete all'interno di un network aziendale è una configurazione molto diffusa; così come nel caso di file e cartelle condivise, la possibilità di inviare input di stampa da qualsiasi postazione facente parte la rete costituisce ormai una realtà ed è diventato uno strumento indispensabile per motivi di comodità e risparmi di gestione.

All'interno di reti locali Windows la funzione di server di stampa deve essere svolta tramite una macchina dotata di uno dei sistemi operativi Microsoft di fascia server, come ad esempio Windows 2000 Server o Windows 2003 Server, ovvero tramite Windows XP. Una configurazione frequente per un server di stampa presenta una o più macchine alle quali sono collegate le stampanti che dovranno essere utilizzate da tutte le postazioni. Dal punto di vista di questi ultimi il processo è assolutamente trasparente: dopo il comando di stampa, viene normalmente presentata una lista delle stampanti disponibili, in locale o in rete. Le rappresentazioni di queste e le funzioni relative sono analoghe, sia per le stampanti direttamente collegate alla postazioni sia per quelli disponibili in remoto. Per l'utente sarà sufficiente selezionare la stampante e dare il comando.

La configurazione di sistemi Windows per l'utilizzo di queste funzioni però presenta, a nostro avviso, un notevole svantaggio: tutte le postazioni che dovranno inviare input di stampa devono presentare i *driver* per le stampanti di rete. Le conseguenze di una limitazione di questo tipo sono notevoli e variano dall'impossibilità di stampare tramite computer non configurati *ad hoc* con i driver, ai tempi necessari per configurare inizialmente un'intera rete e per aggiornarli in caso di cambiamento della stampante o dei driver.

Nel primo caso la limitazione colpirebbe, ad esempio, i computer portatili di dipendenti o clienti, sempre più spesso utilizzati all'interno di network aziendali; la soluzione a questo tipo di problema, naturalmente, si "limiterebbe" alla necessità di configurare ogni singola macchina per poter usufruire delle stampanti di rete. Il secondo aspetto, relativo al tempo speso per l'impostazione iniziale delle postazioni e in caso di aggiornamento, non è risolvibile: qualora si volessero sostituire le stampanti di rete sarà opportuno configurare nuovamente ogni singola postazione. Analogamente, in caso di aggiornamento dei driver di stampa dovranno essere aggiornare tutte le postazioni.

A nostro avviso, una configurazione di questo tipo costituisce un notevole spreco di risorse e una notevole limitazione nelle possibilità offerte da un contesto di network.

I sistemi Linux utilizzano comunemente "*Cups*"³¹, il server di stampa più

31 Cfr. www.cups.org

diffuso in ambito Unix: il sistema è facilmente configurabile ed amministrabile tramite la propria interfaccia web ovvero tramite strumenti di amministrazione appositi come “*Webmin*”³². Questo primo aspetto permette all'amministratore di gestire il server di stampa in remoto e risolvere eventuali problemi direttamente dalla propria postazione, consentendo quindi un notevole risparmio in termini di tempo e risorse. Nel caso dei sistemi Windows, come già affrontato nel corso del secondo capitolo, invece, l'amministrazione completa di una macchina può avvenire in remoto esclusivamente tramite interfaccia grafica, utilizzando applicazioni specifiche come *VNC*, rendendo questo tipo di operazioni più onerose.

Un altro aspetto particolarmente importante, nell'ottica dell'utilizzo di server di stampa all'interno di grandi network, è quello di non necessitare dell'installazione di driver su ogni singola postazione che dovrà far uso delle stampanti. La gestione dei comandi di stampa, infatti, avviene in un contesto puramente *server client*: ai client sarà sufficiente inviare l'input di stampa al server disponibile che elaborerà la richiesta e provvederà all'invio del comando alla stampante dedicata. Normalmente, il comando da parte delle singole postazioni avviene tramite “*rlpr*”, la versione di “*lpr*” sviluppata per l'utilizzo in remoto; con questo software, i client devono solo specificare, tramite nome o tramite indirizzo di rete, quale sia il server di stampa destinatario e quale la stampante ad esso collegata.

32 Cfr. www.webmin.org

Un punto debole di *Cups*, però, è la difficoltà nella configurazione del server stesso; l'applicazione presenta, infatti, notevoli possibilità e funzioni, ma deve essere anche opportunamente configurata. La difficoltà in questa fase è estremamente dipendente dalle funzionalità richieste al server di stampa stesso: per configurazioni elementari la procedura di installazione è banale, ma può complicarsi notevolmente in caso di scenari particolarmente complessi e qualora si voglia fare uso delle funzioni più avanzate³³.

L'utilizzo di *Cups* come unico sistema di stampa è possibile solo in contesti di rete caratterizzati dalla presenza esclusiva di postazioni Linux; qualora nel network siano presenti anche postazioni Windows sarà necessario implementare anche Samba, software già affrontato nel corso del presente capitolo³⁴. Questo strumento, oltre a permettere la condivisione tra postazioni Windows e Linux di file e directory, è in grado di gestire stampanti di rete sfruttabili da entrambi i sistemi operativi. La configurazione di Samba per questa funzione avviene in maniera analoga a quanto impostato per file e directory condivise: non sarà quindi necessario lavoro aggiuntivo.

4.3.3 Server DHCP e DNS

Le tecnologie *DHCP* e *DNS* sono frequentemente utilizzate nei network e

³³ Ad esempio, controllo del numero massimo di pagine stampabili da un singolo utente, autenticazione prima della stampa, calcolo statistico sulle richieste fatte dai client.

³⁴ Cfr. § 4.3.1.

in Internet. La prima, “*Dynamic Host Configuration Protocol*”, è di tipo *server client*: il server DHCP all'interno di una rete si occupa del riconoscimento delle macchine connesse, della loro identificazione e dell'assegnazione di un indirizzo IP di rete. In questo modo, non è necessaria la configurazione di un indirizzo *statico* per ogni singola postazione collegata, ma il server si occupa dell'assegnazione in maniera dinamica. Il protocollo DHCP viene frequentemente utilizzato all'interno di network nei quali vengono spesso collegate macchine non appartenenti alla struttura stessa: biblioteche per i propri utenti, network aziendali per l'utilizzo di computer portatili, punti di accesso wireless *Hot Spot*, ne costituiscono le implementazioni più comuni.

DNS, “*Domain Name System*”, è un sistema tramite il quale effettuare una correlazione tra indirizzi IP (*host names*) e nomi di dominio (*domain names*): con questa “traduzione” di numeri in caratteri alfanumerici, un server DNS permette agli utenti di una rete, ad esempio in Internet, di raggiungere una risorsa online digitando semplicemente “*http://www.unifi.it*” piuttosto che “*http://150.217.1.40*”, che indica l'indirizzo reale del server web dell'Università di Firenze. *DNS* è una tecnologia alla base di Internet stessa e, per quanto non se ne conosca spesso la presenza, indispensabile per l'utilizzo di una rete vasta³⁵.

Entrambi gli strumenti, DHCP e DNS, agiscono in maniera assolutamente

³⁵ Per una persona, infatti, sarebbe impensabile ricordare indirizzi IP a memoria al fine di contattate determinati siti web; la tecnologia DNS ci permette, invece, di ricordarci semplici nomi corrispondenti all'indirizzo del sito.

trasparente all'interno di un network e la connessione di macchine Windows, Linux o di altri sistemi operativi non generano alcuna difficoltà nel riconoscimento delle connessioni e nell'utilizzo dei servizi. In sede di migrazione verso sistemi Linux, quindi, non si incontreranno particolari difficoltà nel permettere a tutte le postazioni lo sfruttamento di entrambe le tecnologie.

4.3.4 Proxy Server

Il *Proxy server* è un servizio di rete tramite il quale i client possono compiere connessioni di rete indirette verso risorse esterne. Il meccanismo è molto semplice: la macchina client si collega al server proxy e fa richiesta per una determinata risorsa, sia essa una pagina internet, il collegamento ad un server di posta o l'utilizzo di un determinato protocollo di rete. Il proxy a questo punto effettua la richiesta e trasmette il risultato al client. Una delle implementazioni più frequenti è quella di *Web Proxy*, spesso utilizzata per velocizzare la navigazione di determinati siti web (tramite un meccanismo di *cache*³⁶), e non causare un inutile sovraccarico di banda in entrata, ovvero per motivi di sicurezza.

Per motivi di efficienza, il server proxy si occupa della memorizzazione, a

³⁶ In ambito informatico, la *cache* è un insieme di dati, ottenuta tramite copia di dati originariamente accessibili da un altro punto. Questo sistema è utilizzato normalmente per evitare inefficienze (nella velocità di acquisizione dei dati, nei tempi di accesso) e sovraccarichi di sistema. I contesti in cui viene utilizzata la cache variano da sistemi hardware a software: al primo caso appartengono, ad esempio, la cache delle CPU e dei dispositivi di memorizzazione.

determinati intervalli di tempo configurabili, di siti web o risorse rese disponibili *online*; quando una postazione client fa richiesta di queste informazioni, riceverà i dati memorizzati nella cache del proxy e non dal sito internet originario. Configurazioni di questo tipo sono particolarmente utili quando molti client di una rete facciano frequente uso di risorse online prevedibili a priori; in questo modo, infatti, le connessioni saranno molto più rapide ed efficienti (la connessione avviene in locale tra client e proxy) e la connessione internet risulterebbe disponibile per le altre postazioni.

Per motivi di sicurezza, la *ratio* dell'utilizzo di un server proxy consiste, invece, nel fatto che i client si connettano ad Internet e ricevano dati “potenzialmente” pericolosi in maniera *indiretta*, tramite la mediazione del server. Accentrando le comunicazioni in questo modo³⁷, è possibile anche effettuare un filtraggio dei dati tramite software appositi: le implementazioni in questo caso possono variare dalla semplice scansione del traffico dati alla ricerca di malware³⁸ al blocco della possibilità di visualizzare determinate risorse online³⁹ o, ancora, al controllo della posta elettronica al fine di “eliminare” le email spam⁴⁰.

Così come nel caso di DHCP e DNS affrontati nel paragrafo precedente, il funzionamento del server proxy all'interno di un network è assolutamente

37 Un meccanismo simile è rappresentato dal *gateway*; anche in questo caso, infatti, il traffico di dati avviene tramite una singola macchina. A differenza del proxy, però, le richieste non sono mediate, ma sono effettuate direttamente dalle singole postazioni client.

38 Cfr. www.squid.org e www.clamav.org. I due strumenti open source, “*Squid proxy server*” e “*Clamav antivirus*”, se usati insieme possono svolgere questo tipo di controllo.

39 Funzione elementare di numerosi proxy server.

40 Questo tipo di controllo è effettuabile abbinando software *ad hoc*, come *Spamassassin*, ad un proxy. Cfr. www.spamassassin.org

trasparente dal punto di vista dei client: le connessioni di questi ultimi, infatti, non dipendono dal sistema operativo utilizzato o dalle singole applicazioni.

Analogamente, qualora si utilizzasse un sistema Windows come proxy server, come “*WinGate Proxy Server*” o “*CCProxy*⁴¹”, entrambi software commerciali, i client Linux non avranno difficoltà ad utilizzare il servizio di rete al termine della migrazione; se si optasse, invece, per l'implementazione di Linux anche per i server proxy, la soluzione software più valida è a nostro avviso costituita da *Squid proxy server*, applicazione distribuita con licenza GPL e liberamente disponibile online.

4.4 Processo di migrazione pilota

Il processo di migrazione implica un cambiamento profondo che colpisce l'intero network aziendale: gli accorgimenti da rispettare per rendere il passaggio a Linux più efficiente e meno traumatico sono diversi e sono stati affrontati nel corso del nostro lavoro. Molto probabilmente si incontreranno difficoltà, resistenze ed incertezze al “cambiamento”: di queste, alcune potranno essere previste a tempo debito e “risolte”; altre, invece, dovranno essere affrontate al momento e superate rapidamente, arrecando il minor numero possibile di rallentamenti alle normali attività.

Per effettuare una migrazione di tutte le postazioni all'interno di un'azienda,

⁴¹ Cfr. rispettivamente www.wingate.com e www.youngzsoft.net/ccproxy/

è necessario considerare attentamente il rapporto con gli utenti finali dei client, dando adeguato peso al “fattore umano”. Le difficoltà relative non si possono circoscrivere esclusivamente ad aspetti tecnici, ma saranno anche rappresentate dalle reazioni delle persone e dal loro atteggiamento; non prestare l'adeguata considerazione a fattori di questo tipo, non rappresenterebbe solo un errore, ma anche un importante rischio di insuccesso.

Nei paragrafi seguenti illustreremo la fase iniziale della migrazione, grazie alla quale sarà possibile comprendere quale possa essere la portata dell'implementazione di Linux e prevedere approcci e comportamenti utili nella migrazione, ed, infine, le ultime fasi del processo di implementazione di Linux.

Il primo passo da compiere è, a nostro avviso, quello di procedere con una *migrazione pilota* prima di intervenire sulla totalità delle postazioni. In questo modo, si possono ottenere importantissime informazioni e spunti per il miglioramento, dei quali usufruire in sede di migrazione completa. Dopo aver raccolto *feedback* di questo tipo, sarà possibile rivedere determinati aspetti del processo, correggere eventuali disfunzioni e provvedere alla realizzazione della migrazione completa.

La migrazione pilota può interessare una piccola frazione del totale dei client, calcolati in base alla dimensione del network. Sarebbe possibile scegliere quali tra queste ultime coinvolgere nella prima fase tramite

adesione volontaria ovvero in altro modo. La prima soluzione probabilmente porterebbe ad un miglior feedback, ad un approccio più costruttivo e ad una maggiore motivazione dei partecipanti. Se si dovesse, invece, procedere con la scelta delle postazioni da interessare, sarebbe, a nostro avviso, opportuno selezionare i soggetti non solo tra gli utenti con maggiori esperienza in campo informatico, ma anche tra quelli “meno esperti”. Un altro fattore da considerare potrebbe essere l'età degli utenti, prediligendo soggetti più giovani che potrebbero presentare un maggior interesse, o una minor avversione, verso la *novità*.

Una conseguenza, infatti, che non è possibile eliminare, sarà la percezione del cambiamento che *tutti* gli utenti delle postazioni inevitabilmente avranno: sarà possibile mitigarlo, ma non annullarlo. E' quindi opportuno preparare adeguatamente i futuri utenti finali, motivando determinate scelte e presentando, con anticipo, alcune situazioni che si potranno incontrare in seguito alla migrazione. Avere un ristretto numero di soggetti che intervengono “in fase di sperimentazione” permette alla direzione di raccogliere informazioni utili per potersi preparare al passo successivo.

Gli elementi più importanti della migrazione pilota sono:

- Applicazioni
- Sistema operativo
- Infrastruttura

- Raccolta del feedback

4.4.1 Applicazioni

Una volta selezionati gli utenti per la migrazione pilota è necessario cominciare a valutare l'impatto del cambiamento delle applicazioni utilizzate; l'implementazione di Linux come sistema operativo, infatti, non riteniamo essere per gli utenti finali l'ostacolo più difficile della migrazione. Il punto dal quale cominciare è di prevedere l'utilizzo delle “nuove” applicazioni partendo, dove possibile, dalla versione per sistema operativo Windows. Molte delle applicazioni che abbiamo proposto nel corso di questo capitolo, infatti, oltre ad essere disponibili nella versione Linux, sono presenti anche per i sistemi Microsoft: è questo il caso di *Openoffice*, *Mozilla Firefox*, *Thunderbird* per la posta elettronica, *the Gimp* per il fotoritocco e la grafica professionale, *Mozilla Sunbird* per la gestione dell'agenda e dei contatti. In questo modo, infatti, l'utente avrà la possibilità di abituarsi per gradi e più lentamente ai nuovi software; sostituendo poi le singole applicazioni non contemporaneamente, ma a debita distanza di tempo, sarà anche possibile per l'utente prendere dimestichezza con le singole applicazioni in maniera più approfondita.

Un approccio di questo tipo permette anche ai soggetti che dovranno prestare assistenza tecnica di concentrarsi su singole aree di intervento che,

molto probabilmente, saranno comuni a tutti gli utenti partecipanti alla migrazione pilota. La maggior parte di essi, infatti, presenterà dubbi e domande in merito alle nuove interfacce delle applicazioni, alle nuove procedure per svolgere determinati comandi e, in definitiva, a questioni legate al cambiamento delle sole applicazioni. Limitando il raggio di intervento del servizio tecnico diventa anche più semplice procedere alla redazione di particolari documenti e risorse per la documentazione che possono rivelarsi fondamentali in sede di migrazione completa per poter assistere anche gli altri utenti.

4.4.2 Sistema operativo

Dopo un primo periodo di introduzione all'utilizzo delle nuove applicazioni si dovrà procedere con la sostituzione del sistema operativo installando Linux.

Dal punto di vista tecnico, proponiamo l'utilizzo di particolari software specifici per l'installazione di Linux su diverse postazioni nello stesso momento che si riveleranno ancora più importanti in sede di migrazione completa; in questa fase preliminare, però, si potrebbe fare un primo tentativo di implementazione di applicazioni come *“FAI-Fully Automated Installation”*, progetto sviluppato dal Dipartimento di Informatica dell'Università di Köln, o l'analogo *“NAIS”*⁴².

⁴² Cfr. rispettivamente <http://www.informatik.uni-koeln.de/fai/> e <http://nais.sourceforge.net/>

Utilizzando simili software è possibile procedere con la configurazione delle impostazioni del sistema operativo, scelta del software da installare, delle regole e dei criteri di installazione su una singola macchina e, una volta memorizzate queste informazioni, replicare le installazioni contemporaneamente su decine o centinaia di altre postazioni, senza dover eseguire nuovamente operazioni ripetitive e gestendo l'intero processo in remoto. Così facendo è possibile nel volgere di breve tempo installare e configurare un sistema Linux funzionante in diverse postazioni anziché impiegare ore, o giorni nella peggiore delle ipotesi, per la stessa operazione, sprecando tempo e risorse⁴³.

Una volta affrontato il tema dell'installazione del sistema operativo e della sua configurazione, sarà necessario ancora una volta assistere gli utenti della migrazione pilota con le difficoltà ed i problemi che potrebbero incontrare a causa del cambiamento. Molto probabilmente questa fase costituirà anche il “termometro” della nostra migrazione, una buona approssimazione di quello che sarà il risultato dell'intero processo, una volta concluso.

⁴³ I software consigliati sono stati sviluppati principalmente per l'installazione di sistemi *Debian GNU/Linux*, per quanto sia comunque possibile scegliere distribuzioni diverse. Il tempo necessario per l'operazione può variare sensibilmente in base alla quantità di software aggiuntivo da installare sulle diverse postazioni e dal collegamento Internet utilizzato per farlo: qualora si facesse una copia locale dei pacchetti necessari all'installazione (così come consigliato dagli sviluppatori di “FAI”) e utilizzando personal computer ragionevolmente aggiornati, la procedura di installazione dell'intero sistema può durare pochi minuti.

4.4.3 Infrastruttura

Al fine di procedere con l'installazione di Linux sulle diverse postazioni facenti parte la migrazione pilota, sarà anche opportuno provvedere alla preparazione dell'infrastruttura di rete. Nella fase iniziale del processo, quindi, è necessario affiancare alla preesistente infrastruttura Windows alcune macchine server con Linux: il numero di queste macchine è strettamente legato a quante postazioni sono coinvolte nella migrazione pilota. Per lo svolgimento delle operazioni più comuni sarà sufficiente utilizzare un server (qui inteso con riferimento al software) *NFS* per la condivisione di dati tra le macchine Linux, *Samba* per la stessa funzione tra postazioni Windows e Linux e *Cups* per l'utilizzo di stampanti di rete. Una volta abbandonata questa fase, e completata l'intera migrazione, le macchine server in questione non saranno più necessarie, perché potranno essere sostituite da quelle originariamente presenti.

Può essere inoltre necessario provvedere alla configurazione di determinati servizi in relazione allo scenario provvisorio delineato dalla migrazione pilota, in cui gli ambienti Windows e Linux coesistono. Ad esempio, nel caso di utilizzo di domini Microsoft NT, si dovrà impostare Samba in modo tale da supportare questo tipo di configurazione. Analogamente, se il network aziendale utilizza un dominio di tipo “*Active Directory*”⁴⁴, Samba

⁴⁴ Tecnologia sviluppata da Microsoft nel 1996 ed implementata per la prima volta su Windows 2000; le versioni più recenti di questo protocollo sono alla base della gestione delle directory via rete di Windows Server 2003.

dovrà essere integrato con Kerberos⁴⁵ al fine di permettere l'autenticazione con server Windows di questo tipo.

4.4.4 Terminal Server

Come già esposto nel corso del capitolo⁴⁶, in particolari contesti lavorativi può accadere che la migrazione di alcuni software sotto Linux non sia fattibile. Particolari applicazioni possono non essere disponibili per il sistema operativo libero ovvero le funzionalità offerte non siano paragonabili alla versione originale. Abbiamo illustrato le tecnologie che possono essere sfruttate al fine di permettere l'esecuzione di applicazioni Windows sotto Linux, come nel caso di VmWare o Wine per l'emulazione delle API; un altro strumento utilizzabile per migrare le applicazioni problematiche, anche questo illustrato nel quarto capitolo⁴⁷, è Terminal Server. Con questo approccio, i software in questione sono installati su uno o più server centrali, utilizzando sistema operativo Windows, e vengono rese disponibili alle postazioni client. Queste ultime, con Linux, accedrebbero alla macchina centrale con software appositi⁴⁸ e eseguirebbero le applicazioni in remoto.

Nel corso della migrazione pilota, quindi, sarà necessario sperimentare a

45 Cfr. <http://web.mit.edu/kerberos/www/>

46 Cfr. § 4.1.3.

47 Cfr. § 4.1.4.

48 Come, ad esempio, *NX client*. Cfr. § 4.1.4.

sufficienza questo tipo di strumenti per verificare la fattibilità della soluzione tramite Terminal Server, in termini di comodità, facilità d'uso e prestazioni. L'utilizzo di questo tipo di strumenti è molto semplice, ma naturalmente risulterà meno immediato rispetto alla disponibilità locale dell'applicazione; situazioni analoghe si registrano già in alcuni tipi di configurazioni aziendali, nelle quali le singole postazioni devono accedere ad interfacce centralizzate per la gestione di software.. Dal punto di vista tecnico, inoltre, non sono identificabili difficoltà di funzionamento, compatibilità o stabilità⁴⁹.

4.4.5 Raccolta del feedback e assistenza tecnica

L'intera fase della migrazione pilota deve essere accompagnata da un adeguato supporto prestato da parte del servizio tecnico; sarà comunque difficilmente evitabile l'insorgere di difficoltà e resistenze da parte degli utenti finali. Come anticipato, l'esperienza fornita dalla migrazione pilota fornirà, con buona approssimazione, un'idea del risultato finale dell'intero processo; la direzione può, quindi, sfruttare simili *output* per poter migliorare, in corso d'opera, la pianificazione e la realizzazione della migrazione completa. E' fondamentale raccogliere durante la prima fase il maggior numero possibile di informazioni di feedback da parte degli utenti,

⁴⁹ Relativamente all'uso dell'applicazione in remoto. Il ruolo fondamentale nell'interfaccia grafica e nel comportamento, infatti, è giocato dal sistema operativo *host*, ovvero quello installato sulla macchina server, nel nostro caso rappresentato da Windows.

analizzarle e, in base a questo, provvedere ad approfondire gli aspetti relativi e a migliorarne le strategie. Questo tipo di evoluzione nel lavoro potrebbe essere prolungata, rendendo la fase della migrazione pilota più flessibile alle esigenze della realtà operativa, permettendo così un'implementazione della fase finale più semplice ed esente da rischi.

Gli interventi di assistenza prestati agli utenti e le domande da essi poste potrebbero essere raccolte e gestite tramite software appositi⁵⁰, in modo tale da renderne disponibili i risultati agli altri soggetti interessati e agli utenti che interverranno nella fase finale della migrazione. Un approccio simile potrebbe far risparmiare notevoli risorse e tempo al servizio tecnico, risultando, probabilmente, più comodo e piacevole anche per gli stessi utenti.

4.5 Migrazione completa

La fase della migrazione pilota dovrebbe produrre una notevole mole di feedback e di informazioni utili all'implementazione di Linux su tutte le postazioni.

Il passo successivo consiste nel procedere sui restanti computer del network: si dovrà quindi cominciare dall'installazione delle applicazioni in ambiente Windows per un certo periodo di tempo e, successivamente, alla

⁵⁰ In molti software groupware, come ad esempio *eGroupware*, sono disponibili moduli specifici per la raccolta di informazioni sulla "Risoluzione dei Problemi" (*Trouble Ticket System*) e per la creazione di *FAQ*, *Frequently Asked Questions*, lista delle domande poste di frequente.

sostituzione del sistema operativo Microsoft con Linux. Anche in questo caso, è consigliabile per questa fase l'utilizzo degli strumenti software, illustrati in precedenza⁵¹, per l'installazione in serie di diverse postazioni.

Gli utenti protagonisti di questa fase potranno fare ricorso all'intera documentazione raccolta durante la migrazione pilota dove, molto probabilmente, saranno contenute la maggior parte delle domande poste e le difficoltà incontrate con i nuovi sistemi nelle attività quotidiane. Interessando il cambiamento l'intero network aziendale, e quindi la totalità degli utenti delle postazioni, è possibile anche che si sviluppi una sorta di collaborazione tra gli stessi al fine di porgere domande o, addirittura, risolvere difficoltà di altri soggetti. Il processo di migrazione completa poi interesserà anche la parte relativa all'infrastruttura server della rete aziendale, presupponendo l'installazione e la configurazione delle macchine necessarie alla condivisione di file e stampanti, al *web proxy* se presente ed a tutti i restanti servizi resi disponibili dal network. Le macchine server utilizzate, in via temporanea, nel corso della migrazione pilota non saranno più necessarie e potrebbero essere sostituite nell'attività dai server originali. Anche in questa fase, il servizio tecnico può avvalersi di un altro tipo di software, fondamentale per risparmiare tempo e risorse e per prestare un'assistenza più "diretta". Utilizzando software per il controllo remoto del sistema, sia di tipo testuale che di tipo grafico, l'amministratore può rendersi rapidamente conto del problema descritto ed intervenire, o

⁵¹ Cfr. § 4.4.2.

illustrare, direttamente sulla postazione.

Nel primo caso, si può utilizzare una connessione di tipo *SSH*⁵², avendo il pieno controllo dell'intero sistema e potendo provvedere allo svolgimento di determinate operazioni: un esempio pratico di applicazione è rappresentato dall'installazione di particolari software richiesti dall'utente. Qualora infatti si sia optato per non concedere agli utenti comuni i permessi per installare o rimuovere software sul sistema, l'amministratore dovrebbe intervenire direttamente sulle singole postazioni; in questo modo, la procedura è molto più rapida ed effettuabile integralmente in remoto.

La connessione di tipo grafico, invece, permette un approccio diverso al problema: può, infatti, capitare che l'utente si trovi in difficoltà con una determinata applicazione o che non capisca come effettuare un'operazione. In questi casi il servizio tecnico potrebbe utilizzare software per l'esportazione remota dei display, come "*X11vnc*"⁵³: questa applicazione permette la connessione remota dal computer client (ad esempio, quello del servizio tecnico), mostrando su questa il display della macchina server (corrispondente alla postazione dell'utente in difficoltà), ovvero quanto visualizzato sul monitor della postazione: il servizio tecnico potrebbe osservare direttamente quanto descritto dall'utente in difficoltà, limitando gli errori o il tempo perso dovuti ad una errata comprensione del problema. L'amministratore, in questo modo, può controllare la macchina remota dalla

52 Cfr. § 2.3.

53 Cfr. <http://www.karlrunde.com/x11vnc/>

sua postazione interagendo con il *desktop* della stessa e mostrando, ad esempio, all'utente la corretta procedura da effettuare per svolgere l'operazione richiesta.

L'utilizzo di strumenti di questo tipo possono rendere estremamente più veloce ed efficiente gli interventi, soprattutto in una fase particolarmente delicata e “ricca di richieste” come quella della migrazione completa dei sistemi, consentendo in tal modo una notevole riduzione dei costi in confronto alle pratiche di assistenza tradizionali.

4.5.1 Fattore Umano

Il processo di migrazione completa dei sistemi è un'attività complessa e, coinvolgendo direttamente anche gli utenti finali, può necessitare di una costante evoluzione in corso d'opera. Le linee guida lungo le quali modificare i processi ed implementare determinati strumenti in maniera diversa da quanto inizialmente pianificato non devono essere esclusivamente frutto di ragioni tecniche o contingenti. Dato il coinvolgimento degli utenti finali è necessario stabilire una valida comunicazione con questi, provvedendo, come precedentemente indicato, a raccogliere feedback e spunti di riflessione per poter adattare meglio i sistemi alla realtà operativa. Gli utenti devono essere, a nostro avviso, informati delle ragioni per le quali è stata scelta la migrazione a Linux dei

sistemi, preparandoli anche a tempo debito sulla possibilità di incontrare difficoltà. E' molto importante, infatti, illustrare agli utenti quello che sarà lo scenario iniziale della migrazione per evitare l'assunzione di comportamenti refrattari alla novità che potrebbero portare ad un notevole spreco di risorse o, nella peggiore delle ipotesi, ad un serio rallentamento delle attività. Un atteggiamento positivo nei confronti dell'operazione di migrazione rappresenta un fattore catalizzatore per il successo del processo stesso; parallelamente, un approccio negativo, refrattario e insensibile alle caratteristiche del nuovo contesto, potrebbe creare seri problemi e far correre il rischio di insuccesso.

Il “fattore umano” è complesso e come tale deve essere considerato: a nostro avviso, sarebbe opportuno mostrare agli utenti i vantaggi che potrebbero ottenere *personalmente* dall'adozione di Linux sulle postazioni; mostrare che il *trade-off* tra difficoltà iniziali e vantaggi finali rende il processo di migrazione un importante valore aggiunto per l'intero network aziendale.

Uno degli strumenti che potrebbero essere utilizzati per introdurre gli utenti al nuovo sistema è rappresentata dalle cosiddette “*Distribuzioni Live*”. Questo tipo di prodotti, normalmente disponibili su un supporto CD o DVD, contengono l'intero sistema operativo Linux e una serie molto ricca di software a corredo, dall'elaboratore di testi alle applicazioni di fotoritocco, dal browser web alle applicazioni di posta elettronica. Il vantaggio notevole di queste distribuzioni, in confronto con i normali CD di installazione, è quello di eseguire tutto all'interno della memoria RAM, non necessitando, quindi, la scrittura su disco rigido e la modifica dei dati

esistenti. Sono spesso utilizzate all'interno di fiere per mostrare le potenzialità di determinati prodotti software; sono anche consigliate agli utenti interessati a Linux che desiderano prendere dimestichezza con il nuovo sistema prima di procedere con l'installazione dello stesso. Avviando il computer con una distribuzione Live, infatti, non viene modificato in nessun modo il contenuto degli *hard disk*, pur potendo accedere ai dati presenti sugli stessi; tutta la dotazione software di queste distribuzioni viene eseguita all'avvio, infatti, e decompressa nella memoria RAM. Un approccio simile causa però un rallentamento delle operazioni rispetto all'esecuzione diretta da disco, ma lo scopo in questa sede non è di rendere immediatamente operative al massimo delle possibilità le postazioni, ma solo di mostrare come potrebbe risultare il sistema in seguito alla migrazione. Gli utenti potrebbero utilizzare i *Live CD* al fine di prendere confidenza con i nuovi applicativi prima della migrazione, in modo da limitare le possibili difficoltà incontrabili.

La prima distribuzione Live sviluppata è stata *Knoppix*, frutto del lavoro di Klaus Knopper, professore universitario di “Sviluppo del software” alla Facoltà di Informatica dell'Università (*Fachhochschule*) di Zweibrücken, Germania. Da questo primo progetto ne sono nati numerosi altri analoghi. Attualmente sono state catalogate 211 Distribuzioni Live⁵⁴, distinte tra loro spesso per gli ambiti di utilizzo, per il focus principale e per la scelta del software a corredo. Sono state così sviluppate distribuzioni specifiche per la produzione audio-video, per *notebook*, per personal computer obsoleti con sistemi ridotti al minimo e così via.

⁵⁴ *Fonte:* Frozentech [Febbraio 2005]. Cfr. <http://www.frozentech.com/content/livecd.php>

All'interno del processo di migrazione la scelta della distribuzione Live da mostrare e fornire agli utenti deve essere a nostro avviso condotta non solo in base alla qualità del prodotto stesso, ma selezionandone anche una che sia quanto più possibile simile ai sistemi che saranno implementati. Se, ad esempio, si sceglierà per le postazioni il ambiente operativo *Gnome*⁵⁵, sarà più opportuna la scelta di distribuzioni che facciano uso di questo ambiente, come *Gnoppix* o *Ubuntu Live*⁵⁶. Discorso analogo vale per la scelta del software a corredo: sarebbe a nostro avviso più opportuno l'utilizzo di una distribuzione che presenti una più ristretta scelta di applicativi per una stessa funzione (preferibilmente uno) in modo tale da limitare il disorientamento iniziale dell'utente.

4.5.2 Formazione

Nonostante l'importante apporto che le distribuzioni Live possono fornire nell'apprendimento dei nuovi sistemi e nel prendere confidenza con Linux, sarà necessario un investimento nella formazione degli utenti. Questa fase potrebbe essere condotta dagli stessi amministratori di sistema, impegnati nella configurazione dei sistemi sin dalle prime fasi della migrazione, o da personale specializzato.

⁵⁵ Cfr. § 4.2.

⁵⁶ Le nostre proposte in questo ambito sono indicative, basate sulla nostra esperienza di utilizzo dei diversi strumenti in questione. La prima distribuzione, ad esempio, è una derivata da *Knoppix* che differisce da questa per l'adozione dell'ambiente desktop *Gnome* piuttosto che *KDE*. La seconda, invece, è una distribuzione molto recente che ha scelto anch'essa *Gnome* come ambiente di riferimento. Per l'amministrazione sarà sicuramente possibile testare altri strumenti e valutare quale sia il più valido. Cfr. rispettivamente www.gnoppix.org e www.ubuntulinux.org

Il momento nel quale condurre la formazione è molto importante: intervenire solo in un secondo tempo, quando il processo di migrazione ha già interessato tutte le postazioni e quando gli utenti hanno già incontrato le prime difficoltà, può essere controproducente, anche a causa dell'approccio negativo che questi possono assumere. E' a nostro avviso più opportuno intervenire preventivamente, piuttosto che essere costretti dalle lamentele, dalle difficoltà e, probabilmente, dai rallentamenti nell'attività lavorativa.

I contenuti e l'approccio alla formazione dovrebbero essere influenzati anche dall'esperienza raccolta grazie alla migrazione pilota; gli utenti interessati in questa prima fase, infatti, possono essere un'utile fonte di informazioni circa le difficoltà riscontrabili nell'utilizzo del nuovo sistema e le impressioni da questo suscitate. Grazie a questo feedback, i formatori possono decidere quale debba essere il percorso d'apprendimento per gli utenti finali, impostando il focus del corso su di un aspetto piuttosto che un altro.

Una prima differenza avvertibile dagli utenti è la “difficoltà” nella personalizzazione del desktop e dell'ambiente operativo della propria postazione: in realtà, su Linux non si tratta di un'operazione più complessa, ma semplicemente *differente*. Le uniche distinzioni rispetto a quanto incontrato con Windows risultano essere nel nome dei menù attivabili con il tasto destro o nell'utilizzo di determinati termini⁵⁷. A nostro avviso, gli argomenti affrontati dalla formazione devono astenersi da tematiche

⁵⁷ Ad esempio, sotto Windows una delle tecniche più rapide per modificare lo sfondo del desktop è di premere il tasto destro sul desktop stesso, scegliere “Proprietà” e poi “Sfondo”. In Linux, invece, con gestore di finestre “XFCE”, è necessario cliccare con il tasto destro, scegliere “Impostazioni” e poi “Sfondo”. Risulta chiaro come le due operazioni non possano essere classificate secondo criteri di semplicità o intuitività: le tecniche sono analoghe, ma fanno uso di termini diversi.

tecniche e poco interessanti per la maggior parte degli utenti, concentrandosi, invece, sull'esposizione dei cambiamenti nello svolgimento delle operazioni più comuni. Condurre il corso con un approccio simile è a nostro avviso un ottimo strumento per stimolare l'interesse dei partecipanti e rendere la formazione più “digeribile” dagli utenti. Durante il corso non deve essere commesso l'errore di sottovalutare o trattare superficialmente questi aspetti, definibili “minori” perché non strettamente legati all'attività lavorativa; l'abitudine all'utilizzo di un sistema porta al disorientamento quando non si è più in grado di svolgere operazioni semplici come quelle indicate in precedenza. L'incapacità di eseguirle, a causa del cambiamento del sistema e delle intrinseche differenze che questo può comportare, può portare all'assunzione da parte dell'utente di un approccio negativo, scarsamente motivato e, soprattutto, poco incline all'apprendimento. Questa è una circostanza da evitare assolutamente, per non correre il rischio che una migrazione, potenzialmente efficace, di successo e ben pianificata, possa incontrare ostacoli e difficoltà.

Nel corso della formazione non bisogna limitarsi, inoltre, all'esposizione dei nuovi sistemi e a questioni strettamente pratiche; è necessario provvedere anche all'esposizione delle ragioni che hanno portato alla scelta di Linux per l'intero network e alla migrazione delle postazioni. Come anticipato precedentemente, infatti, rendere gli utenti partecipi delle motivazioni che hanno portato a questa scelta e consapevoli dei vantaggi che loro stessi potrebbero conseguire può portare ad un approccio più positivo e stimolante nei confronti della migrazione. Allo stesso modo, i formatori dovranno anche curarsi di affrontare le tematiche legate a determinate “opinioni

comuni” su Linux e sul software ad esso dedicato: difficoltà di utilizzo, incompatibilità hardware e disponibilità di software sono quelle più diffuse. Alla luce della rapida evoluzione che il sistema operativo libero ha conosciuto negli ultimi anni, è necessario a riguardo aggiornarsi periodicamente ed informarsi sull'attuale stato dello sviluppo del software. A tal fine, affronteremo in maniera approfondita i principali punti deboli di Linux nel capitolo successivo.

5. Punti deboli di Linux e soluzioni

Linux e il software open source ad esso dedicato costituiscono una realtà molto giovane: il sistema operativo GNU, senza kernel e con le applicazioni a corredo, era già disponibile negli anni Ottanta. L'intervento di Torvalds, però, fu decisivo per il completamento dell'opera, realizzando nel 1991 la prima versione del kernel Linux¹.

Grazie al lavoro svolto da centinaia di sviluppatori di tutte le nazionalità, il sistema operativo libero ha conosciuto un'eccezionale rapidità nell'evoluzione. Il passaggio del kernel dalla serie 2.4 alla successiva 2.6² ha creato enormi potenzialità nel campo della scalabilità di sistema, delle prestazioni, della stabilità, del supporto hardware e delle possibilità in campo di networking. Negli ultimi anni, i cambiamenti sono stati notevoli: molti costruttori di periferiche hardware³ hanno cominciato a prestare supporto per il sistema operativo libero, a sviluppare *driver* specifici, rendendo nella maggior parte dei casi il problema della incompatibilità pressoché totale un ricordo degli anni passati.

Fino a poco tempo fa, infatti, la possibilità di utilizzare periferiche sotto Linux era sostanzialmente concessa o da una sostanziale uniformità nelle

1 Per una più completa esposizione dell'evoluzione del sistema cfr. § 1.2 e § 1.2.1.

2 Le versioni del ramo “stabile” del kernel Linux sono caratterizzate dall'avere il secondo numero di versione *pari* (il cosiddetto *major number*); le versioni “in sviluppo”, invece, presentano solo numeri dispari. Cfr. § 2.1.4.

3 Nel campo delle schede video, ad esempio, *Nvidia* e *Ati*, i due costruttori più importanti, sviluppano da tempo i driver di periferica per Linux. Nel settore delle stampanti, *Epson* e *HP* producono stampanti compatibili con Linux e con il suo server di stampa *Cups*. La maggior parte delle recenti schede di memoria *USB* utilizzano il driver “*usb-storage*” presente nel kernel di Linux dalla versione 2.4 e ulteriormente affinato nel 2.6.

tecniche di costruzione e sviluppo dei componenti (come nel caso di schede audio o lettori CD e DVD) ovvero da team di sviluppatori open source che si dedicavano al *reverse engineering* del dispositivo. Studiando il funzionamento delle periferiche e tentando di comprendere la struttura dei driver per gli altri sistemi operativi, questi si impegnavano nello sviluppo di driver open source, il più delle volte non supportati dalle case costruttrici.

Questa era lo scenario di Linux fino a qualche anno fa; ora rapidamente le cose stanno cambiando. Il sistema operativo ha rapidamente aumentato il proprio bacino d'utenza (nonostante, una percentuale di mercato dei sistemi Windows abbondantemente superiore al 90%), raggiungendo importanti traguardi e stimolando l'attenzione degli addetti ai lavori soprattutto grazie agli importanti investimenti condotti da grandi aziende nel settore⁴. Ciò nonostante la situazione non è ancora paragonabile a quanto offerto agli utenti dei sistemi Windows, incontrastato dominatore del mercato, o anche del sistema Apple; quest'ultimo, pur essendo presente in una ristrettissima nicchia di mercato, è sempre riuscito ad attrarre le attenzioni dei costruttori che, nella maggior parte dei casi, hanno sviluppato driver per il sistema

4 In primis, il riferimento è a *IBM* che negli ultimi anni ha investito importanti risorse su Linux, arrivando anche ad offrire il sistema operativo libero come soluzione software per i propri server. *Novell*, allo stesso modo, dopo aver acquisito l'azienda tedesca *SUSE*, sviluppatrice di una delle più importanti distribuzioni Linux, ha puntato molto su questo sistema. Paradossalmente, la stessa Microsoft ha contribuito a calamitare attenzione sul suo diretto concorrente: pur ignorandolo fino a qualche tempo fa, ha cominciato un'importante campagna di informazione sul confronto tra i propri sistemi e Linux, la rassegna pubblicitaria cosiddetta "*Get The Facts*". *Novell* ha risposto a questo progetto di Microsoft con il suo "*Unbending the Truth: Things Microsoft Hopes You Won't Notice*", campagna di controinformazione sulle presunte inesattezze delle pubblicità Microsoft. Conferma indiretta vi è stata anche tramite l'*ASA* britannico, "*Advertising Standards Authority*", ente a tutela dei consumatori che si occupa delle campagne pubblicitarie, che ha costretto Microsoft a modificare le proprie perché mendaci. Cfr. <http://www.microsoft.com/windowsserversystem/facts/>, <http://www.novell.com/linux/truth/> e www.asa.org.uk

MacOS. A nostro avviso, questo atteggiamento è principalmente legato a due fattori: in primis, la qualità del sistema operativo Apple negli anni precedenti, in confronto a Linux. Quest'ultimo era senza dubbio “primitivo”, complesso e non orientato all'utenza desktop, target di riferimento per i costruttori. In secondo luogo, inoltre, l'assenza di un'azienda, più o meno importante, dietro al sistema operativo libero e la sostanziale diffidenza nei confronti del modello di sviluppo open source, hanno sicuramente giocato un ruolo predominante nel mancato interesse dei costruttori.

In questo capitolo illustreremo i più importanti punti deboli di Linux e chiariremo alcuni aspetti sui quali, spesso, le opinioni sono discordi, principalmente a causa del rapido sviluppo conosciuto negli ultimi anni dal sistema operativo libero.

5.1 Compatibilità hardware

Come affrontato nel paragrafo precedente, la pressoché totale incompatibilità hardware di Linux con le diverse periferiche è un ricordo del passato. Attualmente, la maggior parte delle periferiche e dei componenti sono compatibili e utilizzabili senza problemi; questo è stato possibile grazie al lavoro di diversi sviluppatori, di importanti costruttori che rilasciano driver specifici e grazie all'interesse che grandi aziende (IBM

e Novell in testa) hanno creato intorno al sistema operativo libero. In seguito illustreremo brevemente il livello di compatibilità, distinguendo in base ai diversi componenti e periferiche. Scopo di questa specifica trattazione è di rendere possibile una più completa analisi della propria infrastruttura *prima* di effettuare la migrazione, per meglio considerare le conseguenze di questo processo sul proprio hardware e, se necessario, verificare quale debbano essere le modifiche da apportare.

Un aspetto particolarmente interessante è la *retrocompatibilità* di Linux, assente sia nei sistemi operativi Microsoft che in Apple MacOS. Lo sviluppo del kernel di Linux ha da sempre mantenuto la compatibilità con quanto supportato dalle versioni precedenti; la versione 2.6 del kernel (attualmente la più recente⁵) può essere usata, e sfruttata nelle sue più moderne funzionalità, anche da macchine obsolete. Nei sistemi proprietari, come Microsoft o Apple, tutto ciò non è mai avvenuto: le nuove funzionalità sono prerogative dei sistemi più recenti e, quindi, non disponibili per i sistemi in grado di far funzionare solo le versioni più vecchie ed obsolete. Questa è una delle ragioni che spinge l'utenza dei sistemi operativi proprietari a dover aggiornare periodicamente il software installato per poter sfruttare le nuove funzionalità, affrontando così il costo per l'acquisto di nuove licenze, e spesso a dover addirittura sostituire l'hardware perché non in grado di eseguire le ultime versioni. Discorso analogo vale per l' "obbligo di aggiornamento" causato dall'assenza di

⁵ Al 10 marzo 2005 l'ultima versione stabile è la 2.6.11. *Fonte:* www.kernel.org

supporto della casa sviluppatrice per le versioni più vecchie dei propri prodotti⁶.

Affrontando il tema della compatibilità di Linux con i cosiddetti “integrati”, come schede madri, video, *chipset* e integrati di altro genere, è necessario sottolineare come questi presentino pochi problemi di supporto, se non nulli. Un supporto elementare, infatti, viene prestato alla quasi totalità di questi componenti, anche in assenza di specifico supporto da parte della casa madre. Differente è, invece, il confronto tra le prestazioni e le funzionalità più avanzate offerte dai driver forniti dal costruttore; spesso, come nel caso delle schede video, le differenze sono sensibili, qualora non siano rilasciate le specifiche di sviluppo. Alcuni produttori, *Nvidia* e *Ati* in primo luogo, però, sviluppano da tempo driver di periferica per Linux che forniscono prestazioni paragonabili alle analoghe versioni Windows. Dal punto di vista video, soprattutto, in riferimento ad un utilizzo principalmente desktop e nell'ottica di un'utenza aziendale, il supporto prestato da Linux in questo campo non crea nessun problema di implementabilità. Per le schede madri, analogamente, la compatibilità è ottima e permette di sfruttare al meglio le possibilità offerte dai componenti. Le schede audio e interfacce di rete (*Ethernet* e *Gigabit*) più comuni sono pienamente supportate e funzionanti. Proseguendo nel settore

⁶ Abbiamo affrontato in maniera più approfondita questo punto nel § 2.5 dedicato al “*vendor tie-in*”. Vedremo anche successivamente come evenienze di questo tipo giochino un ruolo fondamentale nella decisione di migrare i propri sistemi a Linux, così come avvenuto nel caso dell'amministrazione comunale di Schwaebisch Hall, affrontato nel capitolo 6. Cfr. § 6.2.1.

networking, anche le moderne estensioni per le connessioni wireless⁷, come quelle offerte dal processore Intel Centrino, sono compatibili con i kernel 2.6, grazie anche al supporto prestato dal costruttore. Nel caso di produttori diversi di chipset, sarà necessario controllare preventivamente la compatibilità.

I *modem* analogici, invece, presentano maggiori difficoltà; negli ultimi anni, infatti, si sono diffusi, soprattutto a causa del basso costo di produzione, i cosiddetti “*software modem*”. Queste sono periferiche, molto simili a semplici schede audio, che non integrano l'hardware necessario alla modulazione e demodulazione del segnale, ma che si limitano ad inviare segnali al sistema operativo che gestisce tutte le operazioni. Molto diffusi nell'ambito dei modem per computer portatili, hanno rapidamente conquistato anche il mercato dei computer desktop. In questo caso, il supporto di Linux per questo tipo di periferica è pressoché *nullo*: fatta eccezione per alcuni produttori di *chipset*, come ad esempio Conexant⁸, la maggior parte non è utilizzabile sotto Linux per assenza di driver. Questa situazione è dipesa principalmente dal fatto che i produttori non hanno mai rilasciato le necessarie specifiche di funzionamento e dalla constatazione di come gli sviluppatori di software libero non si siano eccessivamente interessati di questi componenti. Le conseguenze dell'incompatibilità di Linux con i *software modem* sono, a nostro avviso, ininfluenti nella

7 Protocolli 802.11x.

8 Cfr. www.conexant.com

maggior parte delle possibili configurazioni di un network aziendale, creando, invece, notevoli problemi nell'ottica di un utilizzo casalingo. Per quanto compete il nostro lavoro riteniamo che l'incompatibilità in questione non giochi un ruolo significativo all'interno delle analisi da condurre prima di effettuare la migrazione. Discorso analogo per i *modem xDSL*, largamente supportati dal sistema operativo libero, ma che difficilmente sono implementati all'interno di network aziendali: configurazione tipiche, infatti, prevedono un *modem router* che effettui la connessione con il provider di servizi Internet e poi provveda, tramite cavi, *switch*⁹ e *hub*¹⁰, a distribuire il segnale verso le diverse postazioni. In questo caso, il collegamento tra le macchine e il router avviene con cavo *Ethernet*, o *Gigabit* per le connessioni più veloci, e il sistema operativo utilizzato è assolutamente ininfluenza sulle prestazioni o sul funzionamento del collegamento.

Passando ai dispositivi di memorizzazione esterni, frequentemente collegati al sistema tramite interfaccia *Usb* o *Firewire*, il supporto prestato dalle versioni del kernel 2.6 è ottimo: il modulo del kernel (“driver” per utilizzare una terminologia analoga a quella dei sistemi Windows) è denominato “*usb-storage*” ed è utilizzato dalla maggior parte di questi dispositivi. Per questo tipo di periferica, non è necessario neanche un supporto da parte del

9 Dispositivo utilizzato per connettere differenti segmenti di un network. Si ricorre spesso agli switch al fine di moltiplicare le possibili connessioni ad un network.

10 Dispositivo simile agli switch, dai quali si differenziano per la gestione della banda di collegamento dati: nel primo caso questa viene condivisa da tutti i nodi della rete, mentre nel caso degli switch viene utilizzata per stabilire connessioni dedicate tra nodi individuali.

costruttore perché i diversi prodotti hanno affinità costruttive notevoli e differiscono solo lievemente l'uno dall'altro.

Nel caso, invece, delle stampanti, lo scenario è molto diverso: la compatibilità del singolo modello deve essere *controllata preventivamente* e verificata *caso per caso*. Alcuni costruttori, infatti, sviluppano driver specifici per Linux, ma talvolta alcuni modelli della stessa azienda non sono supportati o presentano notevole carenza di funzionalità avanzate offerte dai driver. Stampanti commercialmente simili, come linea di prodotto, costruttore o funzionalità, possono presentare un grado di supporto sensibilmente differente e *non prevedibile* a priori. Alcune aziende, tra le quali *Epson* e *HP*, si sono impegnate nello sviluppo di driver di ottima qualità, ma questo settore rimane ancora un ostacolo importante per la diffusione del sistema operativo libero e per la sua completa compatibilità. Riteniamo, quindi, che ai fini della migrazione, sia necessaria un'analisi preventiva dei componenti hardware del proprio parco macchine, onde identificare potenziali incompatibilità, valutarne le conseguenze e provvedere con la sostituzione dei componenti, ovvero con soluzioni alternative.

5.2 Difficoltà di installazione, configurazione e manutenzione

Un importante vantaggio dell'utenza Windows è di poter acquistare un

personal computer provvisto di sistema operativo già installato e configurato; con sistemi di questo tipo “pronti all'uso”, l'acquirente, anche se inesperto, può subito dedicarsi all'*utilizzo* del sistema, senza preoccuparsi della sua configurazione. Come descritto in precedenza¹¹, questa strategia della *preinstallazione* del sistema operativo ha portato notevoli vantaggi a Microsoft e all'utenza stessa, rappresentando un importante catalizzatore della diffusione del sistema operativo Windows.

L'utenza *consumer* interessata a Linux non ha mai avuto questa possibilità; solo in tempi recenti, alcune aziende o costruttori di hardware hanno offerto il servizio di preinstallazione e configurazione di una distribuzione a scelta del cliente, frequentemente su computer portatili, ma si tratta di casi molto rari. Come altre volte affermato, l'installazione e la configurazione di un sistema Linux ha rappresentato fino a qualche anno fa un procedimento ricco di difficoltà, di termini tecnici ed incomprensibile agli utenti non esperti.

Gli ultimi tempi hanno però mostrato un'importante evoluzione e miglioramento delle distribuzioni Linux anche su questo aspetto: l'obiettivo non è più attirare soggetti esperti come nuovi utenti, ma colpire il *cuore* del mercato desktop: l'utente comune. Team di sviluppo hanno cominciato a concentrarsi sui programmi di installazione¹² e configurazione iniziale del

11 Cfr. § 1.1.

12 Strumenti utilizzati per installare il sistema operativo e normalmente eseguiti dopo l'avvio del computer da Cd-Rom o DVD. Si occupano di separare il disco rigido nelle diverse partizioni necessarie, nella configurazione preliminare del sistema (componenti hardware, periferiche, impostazioni di rete e altri aspetti) e nell'installazione del software minimo necessario ad avviare la macchina. Nel caso delle distribuzioni Linux, utilizzando esclusivamente software

sistema, raggiungendo importanti traguardi in termini di velocità, semplicità nell'uso ed intuitività.

A nostro avviso, i risultati ottenuti sotto questo aspetto da alcune distribuzioni sono ottimi e permettono anche all'utente non particolarmente esperto di installare un sistema Linux completo e funzionante; sicuramente, tutte le procedure di configurazione specifica dovranno essere curate in un secondo momento e servendosi della documentazione esistente.

Rispondere, però, alla domanda su quale ha, tra Windows e Linux, il programma di installazione più adatto agli utenti inesperti, non è questione facile: la maggior parte degli utenti Windows non ha installato e configurato il proprio sistema, ma hanno avuto la possibilità di acquistare un sistema pronto all'uso. A nostro avviso, un utente inesperto sarà difficilmente in grado di installare un *qualsiasi* sistema operativo.

Ai fini del nostro lavoro, inoltre, è necessario affrontare l'aspetto dell'installazione e configurazione delle macchine, non dal punto di vista dell'utente comune, ma da quello dell'amministratore di sistema; il tempo e le competenze necessarie a svolgere queste funzioni, si ripercuotono inevitabilmente sulle risorse necessarie ad espletarle e, quindi, sui costi della migrazione. Saranno questi i punti sui quali concentreremo la nostra attenzione.

Le procedure di installazione delle più diffuse distribuzioni¹³ sono diventate

libero, i programmi di installazione comprendono anche le applicazioni specifiche, come *Openoffice*, *the Gimp*, e altri applicativi che, in ambito Windows, sono normalmente installati utilizzando altri supporti e software forniti da terze parti.

13 Citiamo, ad esempio, *Debian GNU/Linux*, *Ubuntu Linux*, *Suse*, *Fedora Core*, distribuzione

affrontabili da qualsiasi soggetto con una discreta esperienza in ambito informatico. Questo tipo di strumenti, inoltre, garantiscono nella maggior parte dei casi tempi di installazione molto limitati rispetto a quanto richiesto con i sistemi Microsoft. Gli strumenti software per l'installazione in serie di sistemi Linux¹⁴, inoltre, assenti in ambito Windows, permettono di velocizzare enormemente i tempi necessari a svolgere questa operazione su una moltitudine di macchine. In questo caso, poi, l'assenza di driver di terze parti necessari al sistema operativo libero¹⁵, permette una notevole riduzione dei tempi di installazione; allo stesso modo, l'inclusione della maggior parte dei software specifici all'interno delle distribuzioni stesse consente all'amministratore di sistema di installare in un unico momento sistema operativo e applicazioni terze.

In questo caso i sistemi Windows non offrono le stesse funzionalità: gli applicativi specifici, normalmente, sono di tipo commerciale e distribuiti con supporti separati, necessitando, quindi, dell'installazione manuale in un secondo momento. Una possibilità concreta per velocizzare queste operazioni sui sistemi Microsoft è di fare uso di software di *disk cloning*¹⁶: questa soluzione, però, è percorribile solo qualora le macchine da

nata da *Red Hat Software*, e *Mandrake*.

14 Come FAI, Fully Automated Installation. Ne abbiamo parlato più approfonditamente nel § 4.4.1.

15 Su Linux, infatti, i driver per la gestione della maggior parte dei componenti e periferiche sono gestiti direttamente dal kernel e inseriti al suo interno. Il sistema all'avvio rileva le caratteristiche hardware della macchina e carica dinamicamente in memoria i moduli (driver) del kernel necessari.

16 Questi strumenti eseguono delle copie immagine fedeli di quanto contenuto all'interno di un disco rigido o di una sua partizione; ripristinando queste immagini è possibile velocizzare notevolmente le operazioni di installazione e configurazione di una macchina.

configurare abbiano le stesse caratteristiche hardware, altrimenti sarà impossibile svolgere tutte le operazioni in un unico momento¹⁷.

Riteniamo che le operazioni di installazione di Linux, soprattutto su sistemi simili, tipica configurazione di network aziendali, sia un'operazione semplice e ragionevolmente rapida, soprattutto in confronto al tempo necessario per le stesse operazioni svolte su sistemi Windows.

L'aspetto della configurazione dei sistemi, qui intesa come personalizzazione, impostazione del sistema operativo e dei software, può presentare, invece, maggiori difficoltà rispetto a quanto offerto dai sistemi Microsoft. La configurazione dei software non richiede particolari accorgimenti tecnici o competenze, ma avviene normalmente in maniera diversa da quanto incontrato nei sistemi Windows e, permettendo una notevole personalizzazione, può necessitare di tempi più lunghi e di maggior attenzione. Un vantaggio notevole è, però, costituito dalla rapidità e facilità di trasferimento delle impostazioni dello stesso software da una postazione all'altra: una volta eseguito su una macchina, è possibile replicarlo sulle altre postazioni tramite una semplice copia di file e directory. Questo avviene grazie all'assenza in Linux di un modulo analogo al "Registro" di Windows, al quale spesso le applicazioni terze fanno ricorso per memorizzare numerose impostazioni, e grazie al fatto che la

¹⁷ Facendo una copia *statica* del contenuto del disco, i software di disk cloning ripristineranno anche l'intero sistema operativo, compreso di driver e software aggiuntivo. Utilizzare su una macchina l'immagine creata a partire da un altro sistema, può creare problemi di conflitto tra i driver installati e l'hardware presente sulla macchina. Analogamente, ci saranno difficoltà qualora sulle due postazioni siano presenti dischi rigidi con differenti caratteristiche.

configurazione dei software avviene in maniera trasparente e facilmente “tracciabile”¹⁸.

In definitiva, quindi, la configurazione software di una singola macchina può richiedere tempi notevolmente superiori rispetto a quanto richiesto da un sistema Windows; i maggiori vantaggi di Linux si registrano invece all'aumentare del numero di macchine da configurare in maniera simile. I sistemi Linux sono facilmente replicabili, mentre per il sistema operativo Microsoft è necessario l'intervento manuale e la configurazione su ogni postazione. Il guadagno in termini di tempo e di risorse necessarie al completamento dell'operazione, quindi, dipenderà strettamente dal numero di postazioni da gestire e dalle caratteristiche hardware e di configurazioni delle stesse.

Le differenze tra Windows e Linux sono marcate qualora si proceda con configurazioni approfondite e specifiche *del sistema* per soddisfare particolari esigenze di sicurezza o stabilità dei sistemi; le potenzialità dei sistemi Linux, e *Unix-like* in generale, offerte in questo campo, sono notevoli, ma il tempo necessario alle impostazioni dei permessi, delle utenze, dei gruppi utenti e dell'autenticazione aumenta inevitabilmente,

18 La maggior parte delle configurazioni dei software avviene in Linux tramite semplici file di testo che prevedono determinate direttive e contengono particolari funzioni ed impostazioni. Ad esempio, per le configurazioni *di sistema* delle applicazioni (“*system wide*”, qui intese come le opzioni di comportamento di un determinato software per tutto il sistema; l'utente comune, poi, avrà la possibilità di modificarle a proprio piacimento) i file sono normalmente contenuti all'interno della cartella “*/etc*”, di dominio esclusivo dell'amministratore. I singoli utenti, invece, possono personalizzare il comportamento di un'applicazione tramite modifica dei file contenuti nella propria directory personale corrispondente a “*/home/nome_utente*”. Le impostazioni contenute in questi file avranno effetto *esclusivamente* per l'utente specifico e non per gli altri utenti comuni.

rispetto ad una configurazione più banale. La configurazione, inoltre, richiede agli amministratori di sistema competenze specifiche, rispetto a quanto necessario su sistemi Microsoft, ed in molti casi superiori.

Come specificato in precedenza, il maggior tempo richiesto per questo tipo di configurazione su macchine Linux viene compensato qualora vi sia la necessità di replicare le impostazioni su una moltitudine di postazioni; anche in questo caso, importare questo tipo di personalizzazione su altri sistemi è operazione semplice e veloce.

Il confronto non può avvenire in maniera diretta tra i due sistemi perché le potenzialità offerte sono profondamente diverse tra Windows e Linux; sottolineiamo, però, come a nostro avviso sia sconsigliabile procedere con configurazioni superficiali dei sistemi e riteniamo che una maggiore attenzione su questo aspetto, date le sue implicazioni in ambito di sicurezza informatica, sia indispensabile. Creare sistemi affidabili, stabili e sicuri si può ripercuotere positivamente sul medio lungo termine, evitando, o limitando, l'utilizzo intenso di risorse per interventi di emergenza sulle macchine e per la manutenzione straordinaria.

Un altro aspetto correlato da considerare è relativo alla stabilità offerta dai sistemi Linux nell'ottica delle periodiche reinstallazioni necessarie ai sistemi Windows per mantenere velocità ed efficienza delle macchine. Linux, infatti, non ha bisogno di simili interventi periodici e, in assenza di compromissioni software, come attacchi o infezioni di malware, non

prevede inefficienze legate all'età dell'installazione. Caratteristiche di questo tipo si ripercuotono inevitabilmente sulle risorse da impiegare per la manutenzione dei sistemi, soprattutto nel caso di un parco macchine vasto come in molti network aziendali.

5.3 Difficoltà nell'utilizzo

E' necessario distinguere l'aspetto relativo all'installazione e configurazione dei sistemi da quello del loro utilizzo diretto; nel paragrafo precedente, abbiamo affrontato i primi. Qui di seguito, invece, ci riferiremo alle caratteristiche di Linux e quindi sui possibili effetti della migrazione sull'*utente finale*, colui che effettivamente utilizzerà le postazioni al termine del processo di implementazione. Come affrontato precedentemente, infatti, un'operazione del genere prevede conseguenze inevitabili sui soggetti in questione e trattare superficialmente il fattore umano della migrazione può costituire un grave errore e una potenziale causa di insuccesso¹⁹.

A nostro avviso, le difficoltà nell'utilizzo di Linux, cui spesso si fa riferimento, non sono relative all'*usabilità* di questo sistema e del suo software, quanto piuttosto alle inevitabili differenze riscontrabili rispetto a Windows. La maggior parte degli utenti è abituato ai sistemi Microsoft e si può trovare disorientato di fronte ad un sistema *diverso*, indipendentemente da quale esso sia. Gli stessi sistemi Apple con il sistema operativo MacOS

¹⁹ Alla luce di questo, abbiamo dedicato all'argomento un paragrafo specifico. Cfr. § 4.4.2.

hanno raggiunto importanti traguardi nel campo dell'usabilità, della facilità di utilizzo e intuitività delle operazioni: ciò nonostante un qualsiasi utente Windows, o Linux, si troverebbe inizialmente disorientato, a causa delle differenze riscontrabili.

Valutando Linux nell'ottica dell'usabilità in termini assoluti riscontriamo delle ottima qualità nel sistema operativo libero e un'intuitività notevole, soprattutto per gli “ambienti desktop” (“*desktop environment*”) più evoluti, come Gnome o KDE, per molti aspetti assimilabile a quanto offerto dai sistemi Apple o Windows. Sottolineando ancora una volta come il problema non sia relativo alla complessità intrinseca del sistema ma solo dovuto alla sua diversità e considerando, invece, gli effetti di una migrazione delle postazioni dedicate all'utente finale, riteniamo inevitabili difficoltà e problemi in sede di implementazione. La formazione e la preparazione degli utenti si rivelerà fondamentale e un aspetto da non sottovalutare, così come descritto nel capitolo precedente²⁰.

Nei paragrafi che seguono affronteremo alcuni punti deboli di Linux che interessano l'utilizzo comune del sistema e proporremo degli strumenti e soluzioni per aggirarli in sede di migrazione.

5.3.1 Installazione di applicazioni tramite codice sorgente

Ci sono punti sui quali Linux deve ancora migliorare e che influiscono

²⁰ Cfr. § 4.4.2 “Formazione”.

necessariamente sull'utilizzo del sistema da parte dell'utente comune. Ci riferiamo in primo luogo all'installazione di nuove applicazioni. Come noto, nei sistemi Windows, è normalmente sufficiente eseguire un file di archivio di installazione²¹ che guida l'utente nella procedura, durante la quale vengono richieste una serie di semplici domande sulla cartella di destinazione e su determinate procedure da compiere. Le operazioni da compiere sono banali e note alla maggior parte degli utenti.

Nei sistemi Linux l'installazione di applicazioni è possibile, in linea generale, con due distinte tecniche: utilizzando i pacchetti disponibili per la propria distribuzione²² ovvero compilando il codice sorgente del software ed installandolo manualmente nel sistema²³.

La gestione delle installazioni dei pacchetti avviene in maniera automatica, risolvendo eventuali dipendenze e configurando l'applicazione al termine dell'installazione. Le procedure sono eseguibili completamente via linea di comando, velocizzando notevolmente il tempo necessario alla gestione di diverse postazioni, ovvero tramite software ad interfaccia grafica, gestibili tramite mouse e quindi più adatti all'utente comune. L'approccio necessario è però notevolmente diverso da quanto richiesto in ambiente Windows: sarà necessario confermare l'installazione di eventuali dipendenze software da soddisfare, potranno essere poste alcune domande sulla configurazione

21 Normalmente, un semplice file eseguibile "*Setup.exe*" che al suo interno contiene tutte le informazioni necessarie all'installazione e che, spesso, fa riferimento ad altri file di archivio "*cab*".

22 Ad esempio, i pacchetti *deb* per *Debian GNU/Linux*, gestibili tramite l'applicazione *apt*. Cfr. § 2.3.

23 Cfr. § 2.1.6, per analogia con la compilazione del kernel di Linux.

dell'applicazione, potrebbe essere necessario risolvere eventuali conflitti tra versioni diverse del software o con altre applicazioni. Procedure di questo tipo garantiscono un notevole controllo sul sistema ed una totale trasparenza sulle operazioni svolte, ma possono disorientare utenti non esperti che, molto probabilmente, preferirebbero un approccio più semplice e, se necessario, “meno trasparente”; l'installazione in ambiente Windows, infatti, nasconde completamente all'utente la maggior parte delle operazioni svolte durante il processo, rendendo, da un lato, molto più complesso procedere ad una completa rimozione dell'applicazione ma, dall'altro, estremamente banale completare l'installazione di un software.

Le maggiori difficoltà dell'installazione di software sotto Linux possono non costituire un problema in sede di migrazione qualora si decidesse di non concedere agli utenti comuni i permessi necessari a tal scopo; una gestione simile, da noi consigliata almeno per la fase iniziale della migrazione, assegnerebbe agli amministratori di sistema tale onere. La decisione in merito dovrà, a nostro avviso, essere assunta in base al numero di postazioni del network, alle sue caratteristiche e alla tipologia di utenti delle postazioni e alle loro “competenze”. In caso di un network complesso e con differenti tipologie di utenti, riteniamo che la soluzione più efficiente sia accentrare la gestione in mano agli amministratori di sistema: qualora si dovessero installare delle nuove applicazioni su determinate postazioni, sarà sufficiente collegarsi in remoto alle macchine e dare un semplice

comando per l'installazione, automatizzando, se necessario, il processo. Se il network, invece, è caratterizzato da un numero limitato di postazioni, riteniamo che concedere agli utenti i permessi necessari all'installazione non costituisca un rischio importante.

Il secondo tipo di tecnica per l'installazione delle applicazioni in sistemi Linux, pressoché assente in ambito Windows, è la compilazione del codice sorgente di un software. Si fa normalmente uso di questa procedura quando un determinato applicativo non sia disponibile per la propria distribuzione o per il proprio software gestore di pacchetti. Come ampiamente spiegato nel primo capitolo²⁴, la maggior parte del software per Linux, essendo *free*, rende disponibile il proprio codice sorgente.

Non è possibile però utilizzare direttamente detto codice, ma è necessario procedere alla sua compilazione tramite software apposito. A tal fine, il processo richiede di soddisfare determinate dipendenze che, non essendo gestite in questo caso da un gestore di pacchetti, devono essere risolte manualmente. Le difficoltà e l'attenzione richiesta per svolgere queste operazioni è sensibilmente più complessa di quanto descritto in precedenza.

Un esempio potrà meglio chiarire questo punto. Qualora si volesse compilare dal codice sorgente una semplice applicazione con interfaccia grafica di tipo "*GTK 2*"²⁵ il requisito preliminare sarà quello di utilizzare un software compilatore in grado di trasformare il codice sorgente in codice

²⁴ Cfr. § 1.2, § 1.2.1 e § 1.2.2.

²⁵ Libreria grafica utilizzata e sviluppata principalmente dal progetto *Gnome*, autore dell'omonimo ambiente desktop.

binario, direttamente eseguibile dalla macchina. A tal fine il compilatore necessiterà anche delle librerie *di sviluppo "GTK 2"*: questo tipo di requisito costituisce nella maggior parte dei casi l'ostacolo più importante nella compilazione dai sorgenti di un'applicazione. Le librerie software, infatti, si distinguono in primo luogo tra “semplici” e di “sviluppo”: le prime sono utilizzate per eseguire direttamente codice compilato, mentre le seconde sono *necessarie* alla compilazione del codice sorgente di applicazioni che ne facciano uso. Una volta completato quest'ultimo processo, le librerie in questione non saranno più utilizzate per l'esecuzione dell'applicazione compilata. Questa è la differenza principale che distingue l'utilizzo delle librerie da parte di software “binario” e software da compilare dal codice sorgente. Nell'esempio sopra menzionato, la nostra applicazione presentava come requisito di dipendenza la presenza delle sole librerie di sviluppo *"GTK 2"*, ma nella maggior parte dei casi le dipendenze da soddisfare sono molto più numerose, soprattutto per software complessi. Al fine di risolvere *parte* delle difficoltà e degli ostacoli rappresentati da questo tipo di operazioni, possono essere utilizzati strumenti *ad hoc* per la risoluzione delle dipendenze su richiesta del software compilatore. Un esempio è rappresentato, nell'ambito della distribuzione Debian GNU/Linux, da *"auto-apt"*. Quest'applicazione viene eseguita durante la compilazione del codice sorgente e si pone come interprete tra il software compilatore, come *gcc*²⁶, e il gestore di pacchetti di Debian, *apt*. Quando il primo necessita di

26 *GNU C Compiler*, compilatore per linguaggio “C” sviluppato dal progetto GNU.

particolari applicazioni o librerie di sviluppo, *auto-apt* “interroga” *apt* per verificare se la dipendenza possa essere soddisfatta; in caso positivo, quest'ultimo si occupa dell'installazione dei pacchetti necessari. Al termine di questa fase, ripetuta tutte le volte che il compilatore necessita di particolari software, la procedura di compilazione sarà completata e l'applicazione pronta ad essere utilizzata. Il ricorso a strumenti come *auto-apt* può rendere questo tipo di operazioni estremamente più semplici, ma presenta comunque dei limiti notevoli: la procedura di compilazione dura sensibilmente di più rispetto alla tecnica tradizionale e, soprattutto, può non essere efficiente in tutti i casi. Talvolta, infatti, la dipendenza non può essere risolta semplicemente tramite *auto-apt* ed è necessario intervenire manualmente.

Lo scenario qui presentato in merito all'installazione di applicazioni per Linux dovrebbe a nostro avviso sottolineare diversi punti utili all'analisi preliminare e alla migrazione dei sistemi: un primo punto da evidenziare è la notevole difficoltà presentata dall'utilizzo frequente di software disponibile *solo* in codice sorgente. Un'evenienza di questo tipo comporta diverse conseguenze: in primo luogo, l'impossibilità da parte dell'utente comune di procedere con questo tipo di installazione e la necessità, quindi, dell'intervento diretto da parte dell'amministratore di sistema per provvedere. L'intervento di quest'ultimo sarebbe comunque caratterizzato da tempi piuttosto lunghi da dedicare alle singole postazioni. Alla luce di

ciò, riteniamo fondamentale una scelta particolarmente oculata della distribuzione Linux da adottare in sede di migrazione, prestando particolare attenzione al grado di supporto prestato dalla comunità, alla diffusione conosciuta, alla documentazione disponibile e, in ultima analisi, alla disponibilità di software "binario" specifico per la distribuzione in esame, onde evitare di ricorrere frequentemente all'installazione di software da codice sorgente²⁷.

5.3.2 Accesso ai filesystem

Come illustrato in precedenza, i sistemi Linux gestiscono i filesystem in maniera diversa da quanto avviene con Windows. Una prima differenza è avvertibile immediatamente nella diversa denominazione che i due sistemi assegnano ai dispositivi presenti nella macchina: lettere dell'alfabeto per i sistemi operativi Microsoft e distinzioni più specifiche e dai nomi più complessi per Linux²⁸. A riguardo abbiamo già affrontato come questo ostacolo possa essere superato rapidamente tramite la creazione di link simbolici, dal nome simile a quanto incontrato su piattaforma Windows, in modo tale da non disorientare l'utente. Una soluzione di questo tipo, estremamente semplice, rapida ed efficiente, è facilmente applicabile ai

²⁷ *Debian GNU/Linux*, ad esempio, è una distribuzione caratterizzata da ampio supporto e da una notevole ricchezza di software disponibile nel proprio tipo di pacchetti "deb". Altre distribuzioni particolarmente ricche sotto questo punto di vista sono *Mandrake* e *Fedora Core*, entrambe con sistema di pacchetti "rpm".

²⁸ Cfr. § 2.1.3 e § 4.2.2.

dispositivi stabilmente collegati alla macchina e sempre accessibili.

In talune situazioni, però, la particolare gestione dei filesystem in Linux può creare maggiori difficoltà e richiedere interventi più specifici. Per meglio comprendere il punto, affronteremo più approfonditamente il processo di accesso ai dispositivi di memorizzazione su sistemi Linux.

L'intero "albero" del filesystem viene applicato a partire dalla cosiddetta "directory radice" ("*root partition*", indicata con il simbolo della "slash", " / "); in ambiente Windows, potremmo paragonare questo dispositivo con la partizione di sistema "C:". A differenza di altri sistemi, però, Linux gestisce tutte le partizioni ed i dispositivi di memorizzazione a partire da questa directory iniziale. Conseguenza di questa caratteristica è che un lettore DVD, ad esempio, sarà accessibile tramite una directory situata all'interno della *root partition* /²⁹. Al fine di accedere ai dati memorizzati sul supporto, inoltre, sarà necessario "montare" il dispositivo con un semplice comando, indicando il file di dispositivo³⁰. Questo tipo di operazioni, create originariamente per essere utilizzate tramite linea di comando, sono allo stato attuale gestibili direttamente tramite interfaccia grafica che si pone, rispetto al sistema, come interprete dei comandi specificati.

29 Su sistemi Debian, è normalmente */media/dvd* oppure */media/cdrom*. La posizione della cartella può essere personalizzata a piacimento tramite la configurazione di un semplice file.

30 Viene utilizzato a tal fine il comando "*mount*", comune in tutti i sistemi Unix, la cui sintassi prevede di specificare il file di dispositivo (il file associato al dispositivo o alla periferica, presente nella directory */dev/*) e il cosiddetto "*punto di montaggio*" ("*mount point*"). Un esempio potrebbe essere costituito da "*mount /dev/dvd /media/dvd*". Con un simile comando, si indica al sistema di rendere accessibile il contenuto del supporto nel lettore DVD (*/dev/dvd*) tramite la directory */media/dvd*. I file e le cartelle presenti nel lettore DVD saranno quindi navigabili all'interno di quest'ultima cartella.

Nei sistemi Windows, invece, i diversi filesystem sono gestiti tramite lo strumento offerto da “Risorse del Computer”, nel quale vengono mostrate tutte le periferiche di memorizzazione collegate alla macchina.

L'approccio utilizzato nei sistemi Linux può creare difficoltà quando si accede a dispositivi di memorizzazione rimovibili (come ad esempio le *memorie USB rimovibili*). Il problema in questo caso è rappresentato dalla denominazione che il kernel attribuisce ai file di dispositivo rappresentativi della periferica; l'assegnazione avviene in maniera dinamica e non facilmente prevedibile con certezza³¹. Il problema è a nostro avviso non marginale: questo tipo di supporto per la memorizzazione si è ampiamente diffuso grazie alle piccole dimensioni ed al prezzo ridotto. Riteniamo quindi che questo aspetto debba essere affrontato preventivamente, rendendo i sistemi Linux installati nelle postazioni semplici nell'utilizzo e di facile comprensione da parte degli utenti.

La soluzione da noi proposta prevede l'utilizzo di software specifici per la gestione dei file di dispositivo. Questo tipo di approccio è implementabile velocemente su tutte le macchine ed è indipendente dalla distribuzione Linux scelta per le postazioni. Utilizzando interfacce grafiche evolute come

³¹ Per completezza, è necessaria una precisazione. Chiariremo il concetto con un esempio: si prenda il caso di dover utilizzare su una stessa macchina una memoria USB. All'inserimento di questa, il kernel assegnerà un file di dispositivo del tipo */dev/sda* che rappresenterà la periferica. Disinserendola e ricollegandola, a computer acceso, in un secondo momento, il nuovo “nome” assegnato sarà */dev/sdb*. Ripetendo l'operazione, il kernel attribuirà alla stessa memoria USB il file */dev/sdc*. Il lettore comprenderà come l'assegnazione dei file di dispositivo avvenga in ordine sequenziale, presupponendo quindi che l'utente controlli preventivamente il “nome” della periferica assegnato prima di poter accedere ai dati in essa memorizzati. Per quanto l'operazione sia molto semplice (è sufficiente leggere, alla connessione della memoria USB, le ultime righe del file di *log* del kernel), riteniamo che un approccio di questo tipo sia impensabile per un utilizzo strettamente desktop.

Gnome o KDE, ampiamente affrontate in precedenza³², si può fare ricorso a strumenti specifici come “*Gnome Volume Manager*” che si pone come interprete tra le comunicazioni del kernel del sistema operativo e l'utente. All'inserimento di una memoria USB, ad esempio, il software controlla quale sia il file di dispositivo assegnato alla periferica e offre all'utente la possibilità di “montarla” e di navigarne il contenuto. Il sistema può essere configurato affinché l'intero processo avvenga in maniera completamente automatica. Alla luce di quanto esposto, riteniamo che l'utilizzo di software *ad hoc* possa risolvere ampiamente la complessità presentata dalla gestione dei dispositivi rimovibile da parte del kernel Linux³³.

5.4 Scarsità di applicazioni

L'aspetto della disponibilità di software per Linux è da affrontarsi da due diversi punti di vista: da un lato, la disponibilità di *versioni per Linux* di software per Windows e dall'altro di disponibilità di *applicazioni in generale*. La distinzione è importante perché l'analisi richiesta e le

32 Cfr. § 4.2 e § 4.2.1.

33 Quanto da noi proposto non costituisce l'unica soluzione al problema; un approccio diverso, molto comodo in un'ottica di utilizzo personale della macchina, è quello permesso dalla funzione di “assegnazione statica dei nomi” (*persistent naming*) del software *udev*. Tramite questo strumento è possibile definire delle regole che il kernel deve seguire per l'assegnazione dei file di dispositivo: indicando al kernel, ad esempio, delle informazioni caratteristiche della periferica (come il costruttore e il modello), si può imporre la creazione di uno speciale file di dispositivo, noto in precedenza. Non abbiamo considerato la soluzione proposta da *udev* la migliore in sede di implementazione di Linux in un network aziendale, caratterizzato da una moltitudine di utenti e di postazioni, proprio a causa del suo approccio: dovendo definire delle regole in base alle singole periferiche da poter collegare, l'amministratore dovrebbe intervenire ogni qual volta un utente decida di utilizzare un “nuovo” dispositivo. Cfr. <http://www.kernel.org/pub/linux/utils/kernel/hotplug/udev.html>

conclusioni desumibili sono profondamente differenti.

La distribuzione *Debian GNU/Linux*, attualmente, include, gestisce e rende disponibili per l'installazione 16653 pacchetti³⁴ e comprende esclusivamente software libero; un utente di questa distribuzione ha la possibilità di scegliere tra numerose (spesso decine) applicazioni per la posta elettronica, per la navigazione web, per la preparazione di testi, per la programmazione e lo sviluppo di applicazioni e lo svolgimento di altro tipo di operazioni. L'utente di una qualunque distribuzione Linux ha a disposizione una quantità notevole di applicativi tra i quali scegliere quello che meglio si adatti alle sue esigenze e sono, nella maggior parte dei casi, installabili con un semplice comando³⁵. Un esempio lampante di questa ricchezza è costituito dalle quantità di “gestori di finestre” (“*window manager*”) e “ambienti desktop” (“*desktop environment*”) disponibili: questi sono gli strumenti dai quali dipende la gestione e l'aspetto dell'interfaccia grafica sotto Linux. Scegliendo uno piuttosto che un altro, il comportamento e le caratteristiche del sistema possono cambiare sensibilmente e la gestione del sistema essere profondamente diversa. Gli “ambienti desktop” sono applicativi più complessi rispetto ai gestori di finestre: i più importanti, infatti, sono *Gnome*, *XFCE* e *KDE*, mentre i gestori di finestre inclusi nelle maggiori distribuzioni sono sensibilmente più numerosi. Da questo

³⁴ *Fonte*: <http://packages.debian.org> [marzo 2005]. Sottolineamo, però, che il numero di pacchetti non coincide esattamente con le “applicazioni” disponibili, intese come “software applicativi” per lo svolgimento di una particolare funzione. Nel numero indicato, infatti, sono comprese le librerie di sistema, i pacchetti di sviluppo, la documentazione e altri tipi di strumenti.

³⁵ Ci riferiamo in primo luogo alle possibilità offerte dal sistema di pacchettizzazione *apt* di Debian, distribuzione alla quale spesso ci siamo riferiti nel nostro lavoro. Cfr. § 2.3.

esempio risulta in maniera lampante che la disponibilità di software per Linux sia notevole e che, difficilmente, un utente di questo sistema operativo possa incontrare difficoltà a trovare determinati applicativi. Lo scenario, infatti, è analogo per altri tipi di applicazione.

Abbiamo anticipato all'inizio del paragrafo come siano almeno due, però, gli aspetti da considerare in merito alla disponibilità di software per un sistema operativo. Il punto da affrontare successivamente, e forse quello che desta più preoccupazioni in sede di migrazione, è quante siano le versioni per Linux di software Windows. Lo scenario in questo caso è profondamente diverso: le maggiori aziende di software sviluppano da tempo solo per Windows, soprattutto le applicazioni di larga diffusione, e solo in taluni casi distribuiscono prodotti per Apple MacOS. Sotto questo punto di vista, Linux non differisce da Apple: le versioni per Linux di famosi software per Windows sono rare³⁶. La maggior parte degli applicativi “compatibili” sono altamente professionali (come "*Maya 3d*", software di modellazione tridimensionale) e non di larga diffusione. E' indubbio che il dato in questione, relativo alle applicazioni disponibili anche per Linux, sia nettamente a svantaggio del sistema operativo libero. Quanto sopra è riferito esclusivamente al software commerciale, di tipo proprietario: il software open source, invece, come abbiamo visto, trova in Linux la sua stessa origine ed è largamente diffuso per questa e per altre

³⁶ Un'importante distinzione è però costituita da Microsoft Office, disponibile anche per Apple MacOS X.

piattaforme.

Ai fini della migrazione, quindi, sarà necessario principalmente identificare quali, tra i software open source, possa sostituire al meglio le applicazioni utilizzate precedentemente sotto Windows, e con quali risultati; cercare di sostituire software proprietari per il sistema Microsoft con altri per Linux sarebbe da un lato arduo e dall'altro, a nostro avviso, non conveniente sia dal punto di vista economico che da quello di efficienza³⁷. Abbiamo affrontato anche come particolari applicazioni per Windows siano implementabili in altro modo sotto Linux, tramite ad esempio l'emulazione delle API³⁸; questo tipo di strategie, o l'accesso tramite Terminal Server³⁹, potrebbero costituire una valida soluzione, qualora determinati software siano indispensabili per l'espletamento delle attività.

5.5 Interoperabilità con altri sistemi operativi

Linux ha sempre rappresentato una piccola nicchia nel mercato dei personal computer: i suoi utenti erano poco numerosi, il sistema era sconosciuto ai più e difficilmente gli altri utenti o le aziende sviluppatrici si preoccupavano del sistema operativo libero. Per chi crea software per Linux, dedicare particolare attenzione al restante mercato (la cui totalità è rappresentata dai prodotti Microsoft) e sviluppare in modo tale da

37 Cfr. § 2.4 e § 2.5.

38 Ci riferiamo al "Bridging Application", affrontato nel § 4.1.3.

39 Cfr. § 4.1.4.

garantirne l'interoperabilità e la compatibilità è stata da sempre un'esigenza fondamentale. Molti utenti Linux, utilizzando macchine con “*dual boot*”⁴⁰, hanno la necessità concreta di poter accedere a dati memorizzati su partizioni del disco Windows e viceversa. Sono stati così ottenuti importanti risultati nel campo dell'interoperabilità e della compatibilità; questo tipo di risultati non sono assolutamente raggiunti dagli altri prodotti concorrenti di tipo proprietario. Windows, ad esempio, supporta solo i filesystem Microsoft, *FAT* e *NT*; Apple, in maniera analoga, supporta il proprio *HFS*. Linux, invece, gestisce 17 differenti filesystem⁴¹ e può, in questo modo, accedere nella lettura e scrittura di file con numerosi sistemi operativi diversi. Questo lo scenario in relazione alla possibilità di interagire con differenti piattaforme. Lo stesso sistema operativo è disponibile per diverse architetture⁴² che, nel caso di Debian GNU/Linux, sono undici, garantendo una potenziale omogeneità di sistema per differenti macchine.

Un altro punto di vista da analizzare è a nostro avviso relativo alla compatibilità dei più diffusi tipi di file, disponibili anche per altri sistemi operativi. La compatibilità in questo caso non è rappresentata dalla possibilità di accedere ai dati, ma dalla capacità di visualizzarli e

40 Tecnica con la quale vengono installati due o più sistemi operativi sulla stessa macchina. Tramite particolari software viene selezionato all'avvio quale sistema eseguire; qualora si volesse cambiare sistema da utilizzare, è necessario riavviare la macchina e sceglierlo dal menu iniziale. Abbiamo deciso di non proporlo come soluzione implementativa per la migrazione perchè riteniamo questa procedura, ancorchè adatta per un ambito personale, particolarmente lenta e non produttiva per l'utilizzo in azienda.

41 Tra questi, i più importanti sono: *ext2/3*, filesystem nativo per Linux, *FAT* e *NT* di Microsoft, *HFS* di Apple, *ReiserFS*, *XFS*, *JFS*, *SmbFS* e *NFS*, sistemi per le condivisioni di rete.

42 Intendiamo in questo caso le differenze di architettura hardware; i sistemi più diffusi sono basati su processori “*i386*”, ma ne esistono altre. Due esempi sono rappresentati dalle macchine a 64bit (“*ia64*”) ed i sistemi *PowerPC* utilizzati da Apple.

modificarli correttamente. Per condurre un'analisi di questo tipo, onde valutare l'impatto della migrazione su questo aspetto, è necessario identificare quali siano i più diffusi formati dei file utilizzati nella realtà aziendale di riferimento e verificare l'esistenza di software per Linux in grado di gestirli. Una simile ricerca dovrebbe essere condotta nella fase precedente la migrazione, parallelamente all'analisi sulle applicazioni.

In linea generale possiamo comunque affermare che il sistema operativo libero non presenta difficoltà nell'utilizzo dei più diffusi tipi di file per software per Windows o Apple. In campo Office, ad esempio, abbiamo già indicato come la compatibilità dimostrata da Openoffice nei confronti dei documenti creati con la suite Microsoft sia notevole e tale da non creare particolari problemi di interoperabilità. I file “pdf”, utilizzati da software come “*Adobe Acrobat Reader*”, sono perfettamente utilizzabili, anche tramite l'utilizzo della specifica versione per Linux dell'applicazione Windows. I file audio e video sono pienamente supportati, anche nei casi particolari di filmati creati appositamente per sistemi Microsoft; è questo il caso dei file *Windows Media Video (wmv)* che vengono visualizzati senza difficoltà da software come *Mplayer* o *xine*⁴³; analoghi risultati sono registrati in campo grafico con i più diffusi tipi di file⁴⁴.

Alla luce di quanto illustrato, riteniamo che non vi siano in Linux particolari difficoltà nell'interoperabilità o nella compatibilità dei file con altri sistemi operativi, tali da poter rallentare o rendere problematica la

43 Cfr. rispettivamente www.mplayerhq.hu e <http://xinehq.de/>.

44 Ci riferiamo a *gif, jpg/jpeg, png, bmp, tiff*.

migrazione dei sistemi.

6. Un'applicazione pratica: Schwaebisch Hall

Nel corso del nostro lavoro abbiamo spesso sottolineato come la migrazione dei sistemi informatici sia un processo complesso. Coinvolgendo un alto numero di postazioni, utilizzate per differenti scopi e da parte di diversi utenti, tutte le fasi dell'implementazione di Linux in una realtà preesistente necessitano di analisi, pianificazione e considerazioni in merito alle possibili conseguenze. Il nostro lavoro ha più volte indicato quali possano essere le possibili configurazioni, in termini di applicazioni usate, infrastruttura di rete e tipologia di utenza, che un network aziendale può assumere, esemplificando e trattando i casi più importanti.

In questo capitolo affronteremo un'applicazione pratica di quanto da noi trattato in precedenza. La nostra scelta si è rivolta all'esperienza dell'amministrazione comunale di una città tedesca, Schwaebisch Hall.

La scelta è stata condizionata da diverse motivazioni che rendono questo caso particolarmente interessante, come indicheremo nel corso di questo capitolo. Non abbiamo optato per l'applicazione all'interno di un'azienda perché un esempio di questo genere sarebbe stato necessariamente influenzato dalla specificità della realtà cui si riferiva. La configurazione del network di un'azienda, infatti, è legato indissolubilmente al settore in cui opera ed alle esigenze delle singole postazioni; scegliere una simile realtà avrebbe comportato l'esposizione di un insieme di strumenti,

tecnologie, decisioni e soluzioni applicabili in maniera strettamente limitata a configurazioni analoghe. Nel caso di un'amministrazione comunale, invece, interessando la migrazione tutti i dipartimenti, è possibile avere un ventaglio di scenari e di postazioni profondamente diverse le une dalle altre. Per completezza, nel prossimo paragrafo, indicheremo alcuni esempi di aziende private che hanno effettuato, o pianificato, la migrazione dei sistemi su piattaforma Linux; successivamente ci dedicheremo all'approfondimento dell'importante esperienza conosciuta dall'amministrazione comunale di Schwaebisch Hall.

6.1 Applicazioni pratiche nel mondo aziendale

Il mondo aziendale ha mostrato negli ultimi tempi un interesse crescente nei confronti di Linux e del software open source. In un primo momento l'attenzione principale è stata rivolta all'installazione del sistema operativo libero sulle macchine server, allo scopo di potersi avvantaggiare delle caratteristiche di stabilità, sicurezza e flessibilità, nonché dei vantaggi in termini di costo di licenza che il software open source può vantare nei confronti dei prodotti commerciali.

Come più volte sottolineato nel corso del nostro lavoro, un approccio di questo tipo limita i vantaggi conseguiti ad una ridotta frazione dell'intero parco macchine di un network. Una migrazione di questo tipo, però, annulla

anche la maggior parte degli ostacoli derivanti da una migrazione completa: la formazione degli utenti finali delle postazioni, che di fatto non vengono influenzati da un simile cambiamento, la sostituzione delle applicazioni utilizzate sui client e la configurazione delle stesse.

Ciò nonostante, molte aziende hanno deciso, o stanno pianificando, di migrare l'intero network, ivi comprese anche le postazioni degli utenti finali, a Linux ed al software open source. Gli esempi in questo campo sono numerosi e, data la dimensione della maggior parte di queste realtà, possono facilmente mostrare quale sia la validità e l'efficienza mostrata dal sistema operativo libero e dai software ad esso dedicati.

Czech Post, azienda responsabile per il sistema postale della Repubblica Ceca, ad esempio, ha effettuato nel 2004 la migrazione completa a Linux¹. Costituita da più di 3400 uffici dislocati in tutto il paese e 40000 dipendenti, l'azienda ceca è in grado di gestire più di cento milioni di ordini postali all'anno. Affidandosi inizialmente a software proprietari, per la maggior parte sistemi *Windows NT 4*, analizzò l'alternativa offerta da Linux nel momento dell'aggiornamento hardware delle macchine e dell'acquisto delle licenze per le ultime versioni del software². La scelta della distribuzione si è orientata nei confronti di *Suse Linux*, ora gestita da *Novell*, sia per i server che per le 20000 postazioni client, permettendo riduzione dei costi in termini delle licenze d'uso, dell'hardware utilizzato e della manutenzione dei sistemi. Una decisione simile è stata anche intrapresa dal settimo gruppo

1 Cfr. http://www.novell.com/success/czech_post.html

2 Vedremo successivamente come questa fase sia stata il preludio all'analisi sulla fattibilità dell'implementazione di Linux anche nel caso della migrazione dell'amministrazione comunale di Schwaebisch Hall. Cfr. § 6.2.1.

bancario italiano, *BPU Banca*³, che nei prossimi tre anni ha scelto di migrare ottomila stazioni di lavoro e venti server.

I vantaggi in termine di manutenzione dei sistemi sono quelli che hanno influenzato maggiormente anche Craig Manning, *IT Manager* di *Cisco*, azienda leader a livello mondiale nel *networking*, che ha auspicato la migrazione delle macchine desktop a Linux. Egli sostiene, infatti, che la riduzione dei costi di manutenzione raggiungibile costituisca il maggior vantaggio per un'azienda: le doti di flessibilità e versatilità del sistema operativo libero⁴ permettono ad un singolo amministratore di sistema di gestire da duecento a quattrocento macchine in luogo delle quaranta affrontabili con piattaforma Windows⁵.

A livello server, invece, un interessante studio di *Peerstone*, azienda di ricerca statunitense, mostra come le aziende si stiano rivolgendo sempre più a Linux come server per i propri *ERP* aziendali⁶. Secondo lo studio, il sistema operativo libero è accreditato attualmente di una quota del 2% di presenza su questo mercato, prevedendo una crescita fino al 15% nel 2007. Il settore, allo stato attuale dominato da tre aziende sviluppatrici, *SAP*, *Oracle* e *PeopleSoft*, conta un totale di installazioni pari a ottocentomila server in tutto il mondo. Lo studio indica come la crescita dei server Windows e Unix (come *Solaris* di *Sun*) stia rallentando notevolmente negli

3 Cfr. www.bpubanca.it

4 Cfr. nel capitolo 2 § 2.1 e seguenti.

5 Cfr. <http://www.cwi.it/showPage.php?template=articoli&id=12415>

6 *Fonte*: Gould, J., *Will Enterprise Linux Stop Windows?*, Peerstone, 22 Ottobre 2004, disponibile su www.peerstone.com/pdfs/Peerstone_Ent_Linux_to_End_Win_Server_Growth.pdf

ultimi anni, dal 12/15% del 2004 al 2/5% previsto per il 2005. Jeff Gould, autore dello studio per *Peerstone*, afferma come Linux sia diventato la scelta tradizionale per le aziende che hanno deciso di migrare da sistemi Unix, in luogo della piattaforma Microsoft.

6.2 L'amministrazione comunale di Schwaebisch Hall

Schwaebisch Hall è una città tedesca di 36000 abitanti situata nella parte meridionale della Germania, a circa 65km da Stoccarda. L'amministrazione comunale avviene all'interno di diversi edifici, tramite 225 postazioni, le cui attività variano dalla semplice gestione di documenti anagrafici all'organizzazione del *GeoSystem Information* del Comune, presentando necessariamente una notevole varietà nelle mansioni e nelle competenze richieste agli utenti, nonché nelle applicazioni utilizzate. La maggior parte della nostra opera di documentazione è stata possibile tramite il ricorso alla importante mole di documenti che l'amministrazione comunale di Schwaebisch Hall ha reso disponibile⁷ e grazie al supporto prestatoci direttamente.

6.2.1 Nascita del progetto “Linux im Rathaus”

⁷ Bärwolff, M. , Gehring, R. A. , Lutterbec, B., “*Open Source Jahrbuch 2005: Zwischen Softwareentwicklung und Gesellschaftsmodell*”, Berlino, Lehmanns Media, 2005.

Come anticipato, i sistemi informatici prima della migrazione erano costituiti esclusivamente da prodotti Microsoft: le singole postazioni degli utenti finali utilizzavano Windows 98 e Windows 2000, mentre l'infrastruttura era gestita da Windows NT 4.

Nel 2001 Microsoft annunciò che avrebbe terminato il supporto, in termini di aggiornamento, sviluppo e supporto tecnico, per il sistema operativo utilizzato dalle macchine server del Comune; da ciò dipese non solo la necessità di aggiornare l'intero parco software utilizzato sui server, ma provvedere anche all'acquisto di hardware più recente in grado di eseguire le nuove applicazioni. L'amministrazione comunale, grazie all'interessamento del dipartimento IT, valutò le possibili conseguenze che una decisione simile avrebbe comportato, certa della possibilità che annunci di questo tipo potessero in seguito interessare anche altri prodotti utilizzati. Fu così presa la decisione di valutare quali potessero essere le alternative valide alla piattaforma Microsoft, quale il budget necessario ad implementarle, i tempi di realizzazione del progetto e quali le conseguenze. Da queste premesse nacque il progetto "*Linux im Rathaus*"⁸.

6.2.2 Motivazioni

Al termine dell'analisi delle alternative possibili, prevista dall'amministrazione comunale e condotta dal proprio dipartimento IT, è

⁸ "Linux in Municipio"

risultato che Linux ed il software open source fosse l'unica valida piattaforma alternativa. I punti di forza del sistema operativo libero, risultanti dall'indagine, sono stati identificati in:

- Indipendenza
- Sicurezza
- Riduzione dei costi

Il primo punto è relativo principalmente a quello che nel nostro lavoro abbiamo definito “*vendor tie-in*” e che spesso impone all'azienda, utilizzatrice di software proprietario, il cosiddetto “obbligo di aggiornamento”⁹. L'interruzione del supporto per i propri prodotti, annunciato da Microsoft nel 2001, poneva l'amministrazione comunale di Schwaebisch Hall nella necessità, in alcun modo legata ad aspetti tecnici di inadeguatezza dei sistemi, ad effettuare una sorta di “migrazione”: l'aggiornamento della totalità dei sistemi imponeva l'acquisto di nuove licenze e, quindi, il cambiamento di tutte le versioni dei software utilizzati. Nonostante la piattaforma rimanesse legata a prodotti Microsoft, il dipartimento IT era dell'opinione che, data la vastità del processo, questa dovesse essere considerata a tutti gli effetti una “migrazione”. Una simile affermazione viene anche avvalorata dalle notevoli differenze presenti tra le versioni utilizzate all'epoca dall'amministrazione comunale, Windows 98 e

⁹ Cfr. § 2.5.

2000 per i desktop e Windows NT 4 per i server, ed i nuovi prodotti che sarebbero stati implementati dopo l'aggiornamento, principalmente Windows XP e Windows 2000 Server.

Dal punto di vista dei costi, l'analisi stimava in € 250.000 i costi di licenza che l'amministrazione comunale avrebbe dovuto sostenere nei due anni seguenti¹⁰. Fu così valutata l'opportunità di annullare, o quanto meno limitare fortemente, la dipendenza da un singolo fornitore, il cosiddetto “*vendor tie-in*”, tramite l'utilizzo di Linux e del software open source.

Come anticipato, la migrazione all'interno di prodotti Microsoft non imponeva il semplice aggiornamento del sistema operativo e delle applicazioni, ma anche dell'hardware delle singole postazioni. Le esigenze di calcolo richieste dalle ultime versioni del software, infatti, erano notevolmente aumentate, in particolar modo per le macchine desktop; nonostante l'opera di aggiornamento dei sistemi che l'amministrazione comunale aveva svolto periodicamente dal biennio 1997/1998, le richieste dei nuovi prodotti Microsoft erano superiori e necessitavano, quindi, di hardware più recente. Questo fu un altro fattore che portò l'amministrazione a valutare attentamente l'alternativa offerta da Linux e dal software open source.

Come schematizzato in precedenza, l'indipendenza da un singolo fornitore non è stata l'unica motivazione che ha spinto il dipartimento IT a

¹⁰ La stima è stata effettuata in base alle condizioni contrattuali previste tra Microsoft Germania e l'amministrazione comunale di Schwaebisch Hall.

promuovere l'adozione del software open source: il secondo punto affrontato a riguardo è stato la sicurezza dei sistemi. L'aspetto relativo alla protezione dai pericoli diffusi del *malware*¹¹, da attacchi mirati e della sicurezza informatica in generale sono stati particolarmente considerati data l'implementazione in sistemi che gestiscono una grande mole di informazioni sensibili sui cittadini. Le attenzioni principali sono state infatti rivolte alla *protezione dei dati* e delle informazioni memorizzate nei sistemi, al fine di evitare possibili compromissioni e di rispettare le norme relative alla conservazione dei dati sensibili da parte delle pubbliche amministrazioni. Un altro aspetto, dalla rilevanza principalmente tecnica, è costituito dall'*integrità* degli stessi dati: in questo caso il pericolo da affrontare è quello dell'instabilità dei sistemi e dalla loro efficienza, per tutelarsi da possibili danni che ne possano conseguire. In relazione a questo, il dipartimento IT dell'amministrazione comunale di Schwaebisch Hall ha anche valutato gli effetti di una mancata affidabilità dei sistemi: la disponibilità dell'infrastruttura di un'amministrazione di questo tipo deve essere, come condizione primaria, prossima al cento per cento¹².

L'amministrazione comunale, però, non si è limitata all'analisi ed alla ponderazione delle qualità tecniche che la migrazione a Linux dei sistemi

11 Per una più approfondita analisi dei danni prodotti da questo tipo di “applicazioni”, cfr. § 2.1, § 2.1.1 e § 2.1.2,

12 Un indice per misurare la “disponibilità” di un servizio di rete è il cosiddetto *uptime*, che misura il tempo trascorso dall'ultimo riavvio della macchina. Normalmente le macchine server che prestano servizi di una certa importanza presentano indici misurati in anni; un alto valore, ad esempio, mostra una notevole stabilità del sistema, una capacità di aggiornamento del software che non richieda il riavvio della macchina e una garanzia di continuità dei servizi prestati.

avrebbe comportato: è stata data particolare attenzione anche a come le esigenze di affidabilità del servizio, riassumendo in questo modo i punti sopra citati, unite alla necessità di una garanzia assoluta di protezione dei dati sensibili memorizzati nei sistemi, debbano *necessariamente* prevedere *la disponibilità del codice sorgente* delle applicazioni utilizzate. In questo caso il riferimento non si limita al sistema operativo implementato, ma anche alle principali applicazioni delle quali l'amministrazione comunale avrebbe fatto uso.

Come anticipato precedentemente, un altro aspetto analizzato dal progetto “Linux im Rathaus” è stato la riduzione dei costi possibile grazie all'utilizzo del software open source. A riguardo, l'amministrazione comunale aveva avuto modo di conoscerne gli effetti in seguito all'implementazione di Linux su alcuni server avvenuta nel 1997. I campi di applicazione furono, inizialmente, la gestione del *DHCP*¹³, del *Domain Name Server*¹⁴, della posta elettronica interna e del *mail-relay*¹⁵: le macchine server che prestavano questi servizi, quindi, da tempo utilizzavano software libero. I passi successivi si concretizzarono con la gestione delle condivisioni di filesystem via rete¹⁶ e per la condivisione del sistema di stampa¹⁷. L'amministrazione comunale nel 1997 non aveva ancora ritenuto opportuno

13 Cfr. a riguardo § 4.3.3.

14 Cfr. a riguardo § 4.3.3.

15 Il servizio in questo caso si occupa di inoltrare determinate email da un indirizzo ad un altro, all'interno della rete dell'amministrazione comunale.

16 Il dipartimento IT per assolvere questa funzione fece uso del noto software *Samba*. Cfr. a riguardo § 4.3.1.

17 I server di stampa furono gestiti tramite il software CUPS. Cfr. a riguardo § 4.3.2.

l'utilizzo di Linux per la memorizzazione dei dati sui database server.

L'esperienza maturata nel quadriennio 1997-2001, riportata dal progetto “Linux im Rathaus”, mostrò alcuni importanti aspetti che hanno giocato un ruolo fondamentale nella decisione di effettuare la migrazione completa dei sistemi a Linux. In primo luogo, non fu ritenuto necessario l'aggiornamento dell'hardware delle diverse macchine server utilizzate: le esigenze in termini di capacità di calcolo e di memoria richieste da Linux erano perfettamente in grado di essere soddisfatte con le macchine già in possesso: il cosiddetto “obbligo di aggiornamento” è stato, quindi, considerato non come un'esigenza tecnica, ma come un obbligo imposto dall'eventuale migrazione dei sistemi alle ultime versioni dei prodotti Microsoft. Il risparmio conseguibile dall'implementazione di Linux sulle sole macchine server fu analizzato ed applicato, analogamente, alle postazioni desktop degli utenti finali: l'alto numero di queste ultime, non poteva che agire come *moltiplicatore dell'effetto*. Le stime condotte dai responsabili del progetto presentavano un risparmio di circa € 500 per ogni singola postazione di lavoro; per mantenere un'alta attendibilità del dato in questione, il dipartimento IT ha deciso non solo di analizzare le postazioni dalle minor esigenze e caratterizzate da una minor specializzazione, ma anche di prevedere l'utilizzo di una distribuzione Linux a pagamento, come *Novell Suse* che prevede dei costi per il supporto e per gli aggiornamenti, senza ricorrere quindi alle distribuzioni liberamente disponibili¹⁸.

¹⁸ A riguardo notiamo come la distribuzione tedesca in questione, *Novell Suse*, goda di

Il dipartimento IT, responsabile del progetto “Linux im Rathaus”, ha anche affrontato l'aspetto dibattuto dei costi da sostenere per la formazione degli utenti, necessaria in seguito alla migrazione a Linux¹⁹. In base alle esperienze maturate in precedenza all'interno dell'amministrazione comunale, in relazione alle “migrazioni” effettuate tra i diversi prodotti Microsoft²⁰, i responsabili hanno ritenuto che i costi della formazione legati a Linux ed altre applicazioni open source sarebbero stati analoghi.

6.2.3 Condizioni da rispettare per la migrazione

Dopo l'esposizione dell'analisi effettuata dai responsabili del progetto “Linux im Rathaus”, e dopo la decisione di intraprendere la migrazione, in accordo con il Sindaco Hermann Josef Pelgrim, furono definite le condizioni che il processo avrebbe dovuto necessariamente rispettare:

- tutti i software utilizzati per le applicazioni di gestione “propria” dell'amministrazione avrebbero dovuto soddisfare integralmente le esigenze delle diverse postazioni, senza arrecare rallentamenti nello svolgimento delle attività;
- le applicazioni “Office” (comprendendo così elaborazione testi, utilizzo di fogli di calcolo, presentazioni) dovevano

un'attenzione particolare in molti casi di migrazione effettuate da realtà operanti in Germania.

19 Cfr. § 4.5.1 e § 4.5.2.

20 Ad esempio, da Windows 3.11 a Windows 98 e da Microsoft Office 95 alla versione 2000.

presentare un grado di compatibilità assoluto; il cambio di applicazioni non avrebbe dovuto inoltre causare alcuna perdita di informazioni rispetto alle risorse precedentemente utilizzate;

- considerando le inevitabili conseguenze sugli utenti finali, la migrazione dei sistemi non avrebbe dovuto comportare in capo a questi ultimi in nessun modo un maggior carico di lavoro rispetto a quanto permesso dalla piattaforma precedente;
- la configurazione dei sistemi, con particolare attenzione nei confronti delle postazioni finali, sarebbe dovuta essere tale da minimizzare il ricorso alla formazione degli utenti;
- il dipartimento IT avrebbe dovuto pianificare ed organizzare tutti i corsi di formazione per gli utenti finali, distinguendoli in gruppi.

I punti sopracitati mostrano chiaramente come l'amministrazione comunale abbia considerato prioritaria la produttività delle postazioni e la minimizzazione dei rischi che avrebbero potuto rallentare o interrompere le normali attività svolte; dello stesso avviso il Sindaco Hermann Josef Pelgrim che spiegò chiaramente che queste condizioni erano assolutamente fondamentali e da rispettare integralmente.

6.2.4 Pianificazione della migrazione

Il progetto “Linux im Rathaus” fu presentato anche in occasione del *Linux World Expo 2001*, svoltosi nello stesso anno a Frankfurt im Rahmen, in Germania. In questa sede, le aziende IBM e Suse si offrirono di prestare supporto all'intero processo.

Nel dicembre 2001 fu deciso che il progetto si sarebbe concluso entro tre anni, determinando un termine finale con la fine del 2004. Il dipartimento IT era dell'opinione che non fosse possibile definire un preciso e rigoroso piano temporale per la migrazione; le difficoltà in merito erano principalmente relative ai sistemi desktop e agli utenti finali delle postazioni, per i quali era stato già prevista la necessità di corsi di formazione. Ciò nonostante si riteneva che il termine dei tre anni potesse essere rispettato senza difficoltà.

La migrazione fu distinta in due importanti fasi:

1. implementazione sui server
2. installazione e configurazione sulle postazioni desktop dei diversi dipartimenti

Scopo dichiarato dell'intera pianificazione è stato anche di creare un importante documento che potesse essere in seguito utilizzato anche da altre

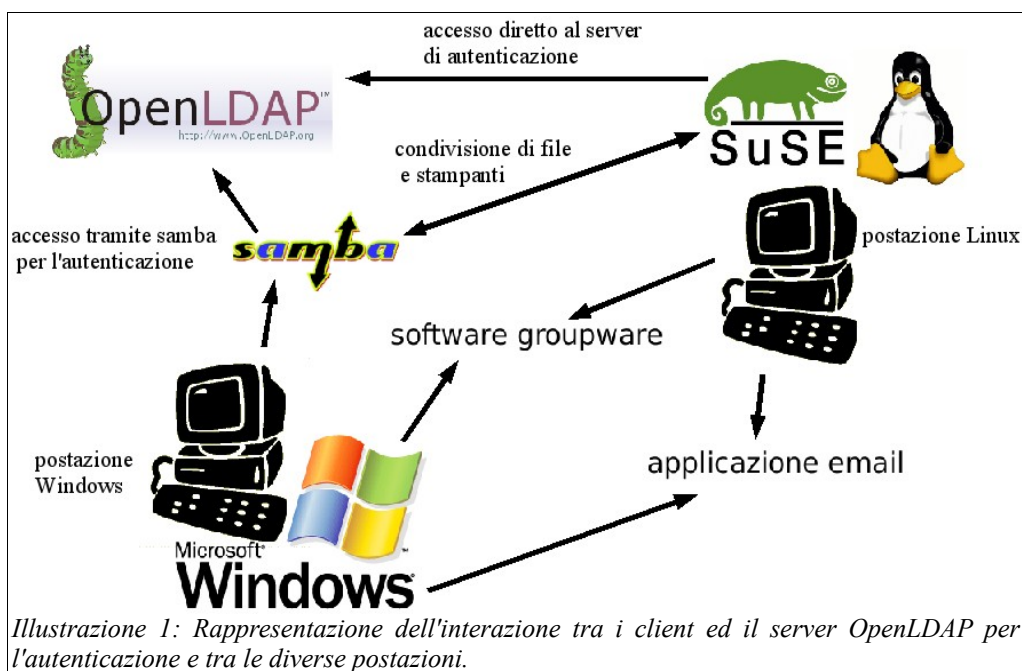
realtà interessate all'implementazione di Linux.

6.2.5 Implementazione tecnica

Il primo passo compiuto dal progetto è stato quello di permettere una gestione centralizzata dell'intera infrastruttura; gli strumenti utilizzati avrebbero dovuto garantire la compatibilità di funzionamento all'interno di una rete mista composta da postazioni Windows e dai sempre più numerosi client Linux. Il dipartimento IT ha optato per *OpenLDAP*²¹; il servizio è stato poi configurato per interagire con il server Samba in modo tale da permettere la comunicazione con i client Windows e permettere a questi ultimi di usufruire del dominio creato come *Primary Domain Controller (PDC)* per l'autenticazione degli utenti. Questo passaggio è necessario per la comunicazione all'interno di una rete mista Windows Linux; questi ultimi client, infatti, possono autenticarsi direttamente sul server OpenLDAP.

L'Illustrazione 1 rappresenta questa interazione.

²¹ *Lightweight Directory Access Protocol*, software open source per la gestione di un dominio di rete. Cfr. <http://www.openldap.org/>



Il dipartimento IT ha deciso di mantenere attivi, senza necessità di aggiornamento, i quattro server IBM *X-Series* precedentemente installati, configurati in modo tale da mantenere sempre attivi, anche in caso di emergenza, i servizi più importanti. La scelta delle distribuzioni si è orientata verso *Suse Linux OpenExchange Server* e *Suse Linux Enterprise Server* per espletare le funzioni di database server, server di stampa, di condivisione file, email ed autenticazione nel dominio.

Dal punto di vista delle postazioni desktop, particolare attenzione è stata rivolta ai software di produttività Office. Tutte le macchine sono state configurate in modo tale da utilizzare OpenOffice per la gestione di questo tipo di operazioni; la suite libera è stata installata su tutte le postazioni,

sostituendo Microsoft Office anche sulle macchine Windows ancora presenti. Questa caratteristica è stata fortemente voluta dal dipartimento IT, identificando in essa un passo fondamentale per il passaggio alla migrazione completa: in questo modo, infatti, gli utenti delle singole postazioni Windows ancora presenti, avrebbero potuto abituarsi più lentamente al cambiamento di applicazione e di sistema operativo. Per quanto concerne la migrazione dei dati, si è optato per una conversione di tutti i dati presenti all'interno di database Microsoft Access²² in database gestiti a livello centrale tramite il sistema *MySQL* e consultabili dalle singole postazioni tramite interfaccia web; per le fasi di transizione, il dipartimento IT ha scelto di usare *CrossOver Office* di *Codeweavers*. Come affrontato nel corso del quarto capitolo²³, questo strumento commerciale permette di utilizzare determinate applicazioni per Windows sotto Linux, specificatamente sviluppato per il supporto di applicazioni professionali, come Adobe Photoshop e Microsoft Office²⁴.

Il progetto “Linux im Rathaus” ha anche introdotto nell'infrastruttura dell'amministrazione comunale di Schwaebisch Hall uno strumento non utilizzato in precedenza: un software di tipo *groupware* gestito a livello centrale da un server ed accessibile dalle postazioni tramite browser. Le

22 Applicazione per una semplice creazione dei database, inserita all'interno della suite Microsoft Office Professional. Utilizza un formato proprietario per la memorizzazione dei dati.

23 Cfr. § 4.1.3.

24 Notiamo come il software di Codeweavers utilizzi per la maggior parte il codice sorgente del programma open source *Wine*, emulatore delle API di Windows. Il lavoro compiuto dall'azienda americana consiste principalmente nella maggior attenzione rivolta al supporto delle applicazioni comunemente utilizzate in ambito aziendale.

motivazioni alla base di questa decisione sono state principalmente quelle di aumentare la sicurezza delle singole postazioni, tramite un accesso centralizzato, e di rendere più efficace la collaborazione e la comunicazione tra i singoli dipartimenti e dipendenti²⁵. La soluzione implementata ha previsto la gestione di un calendario condiviso e consultazione della posta elettronica, accessibile dalle singole postazioni tramite interfaccia web. Lo strumento utilizzato dall'amministrazione comunale è stato sviluppato dal dipartimento IT, in collaborazione diretta con *Suse Linux*.

6.3 La migrazione delle applicazioni specifiche

L'esperienza dell'amministrazione comunale di Schwaebisch Hall è particolarmente indicativa anche per le soluzioni tecniche adottate per risolvere le difficoltà relative alla migrazione delle applicazioni. Nell'analisi preliminare condotta all'interno del progetto "Linux im Rathaus", infatti, è emerso in primo luogo il gran numero di applicazioni utilizzate per lo svolgimento delle operazioni più comuni di amministrazione. Le 84 applicazioni, la maggior parte delle quali sviluppate *ad hoc*, possono essere distinte in tre categorie, così come emerso dall'analisi preliminare:

- applicazioni di tipo *ASP*;

²⁵ Nel corso del nostro lavoro abbiamo affrontato la soluzione rappresentata dalle applicazioni "web based", come i software *groupware*, nel § 3.6 e seguenti, dove abbiamo sottolineato i vantaggi in termini di sicurezza e di riduzione dei costi per l'acquisto delle licenze e indipendenza dal fornitore.

- applicazioni basate su architettura *server client*;
- applicazioni desktop;

La prima categoria è rappresentata da software per il cui utilizzo era necessaria la connessione al server centrale e l'uso, tramite terminale, di applicazioni installate sul server stesso; in questo caso non vi sarebbero state particolari difficoltà connesse alla migrazione, grazie alla notevole ricchezza di strumenti open source per Linux per la connessione via terminale²⁶.

Alla seconda categoria, invece, appartengono le applicazioni che richiedono solo l'interazione con un server centrale che ospita il software da utilizzare; un esempio particolarmente indicativo è rappresentato da tutti gli strumenti con interfaccia web, accessibili semplicemente tramite le applicazioni *browser* come Internet Explorer o Mozilla Firefox. In casi simili, infatti, non vi sono particolari problemi di compatibilità dipendenti dal sistema operativo installato sulle macchine client; questo tipo di applicazioni erano state sviluppate precedentemente dal dipartimento IT, rivolgendo particolare cura alla compatibilità con i più diffusi browser internet.

La sfida più difficile per il progetto “Linux im Rathaus” è stata rappresentata dalla terza categoria di applicazioni: le applicazioni desktop. In questo caso il riferimento è ai numerosi software, installati sulle singole postazioni, utilizzati per lo svolgimento di determinate operazioni di

²⁶ Nella migrazione in esame, ad esempio, è stato utilizzato “x3270”, un emulatore di terminale IBM per *Xfree86*, il server grafico di Linux.

amministrazione. Questa categoria, la numericamente più importante, rappresentava, per la migrazione, la barriera più complessa da superare: la totalità di questi software, infatti, erano stati sviluppati esclusivamente per Windows. La soluzione di utilizzare applicazioni native per Linux non era assolutamente percorribile.

Come affrontato specificatamente nel corso del quarto capitolo, esistono diverse soluzioni adatte ad eseguire applicazioni specifiche per Windows su piattaforma Linux: *Wine* ed il suo derivato *CrossOver Office*, *VmWare* e *Bochs* sono alcuni esempi²⁷. Ciò nonostante, la soluzione offerta da questo tipo di strumenti può non essere efficiente in taluni casi ovvero richiedere una lunga fase di configurazione in altri.

La soluzione scelta dal progetto “Linux im Rathaus” è stata di spostare queste applicazioni, in origine installate su ogni singola postazione, dai client al server: una macchina Windows avrebbe ospitato questi software che sarebbero poi stati utilizzati dai client Linux tramite la tecnologia di Terminal Server²⁸. Questo tipo di implementazione, già affrontata nel corso del nostro lavoro, è stata denominata dai responsabili di progetto “Client Universale”: partendo, infatti, dal presupposto che, all'interno di una rete mista Windows Linux, è pressoché impossibile utilizzare le stesse applicazioni client su ogni postazione, il dipartimento IT ha deciso di risolvere in partenza l'ostacolo. L'approccio scelto, inoltre, presenta

²⁷ Cfr. i paragrafi da 4.1 a 4.1.4.

²⁸ Cfr. § 4.1.4.

vantaggi anche dal punto di vista dell'amministrazione dei sistemi: privando le postazioni client delle applicazioni più importanti, avvicinandosi quindi al concetto di “thin client”²⁹, è possibile diminuire notevolmente gli interventi di manutenzione. Dal punto di vista tecnico la soluzione introdotta all'interno del progetto “Linux im Rathaus” è stata quella di usare un client universale per la connessione al server Windows, utilizzando per motivi di sicurezza una comunicazione cifrata tramite *SSL*³⁰: il software è *NX NoMachine*³¹. Il vantaggio principale derivante dall'utilizzo di questo strumento è la presenza di numerose versioni per i sistemi operativi più diffusi e per altri tipi di dispositivi³².

6.4 Realizzazione della migrazione

Sin dalla nascita del progetto “Linux im Rathaus” è stata manifestata l'intenzione di effettuare la migrazione con una spesa minima.

Per la prima fase del progetto, la cosiddetta “migrazione pilota”³³, sono state scelte 40 postazioni appartenenti a diversi dipartimenti dell'amministrazione comunale. Obiettivo era quello di effettuare la migrazione completa dei client per verificare quali fossero i problemi e le difficoltà non previste dalla pianificazione preliminare. Prima di procedere

29 Cfr. § 3.2, § 3.2.1 e § 3.6 e successivi.

30 *Secure Socket Layer*, protocollo cifrato per le comunicazioni.

31 Cfr. § 4.1.4.

32 Ad esempio, oltre ad essere presente per Windows, Linux e MacOS, gli sviluppatori rilasciano anche client per computer palmari.

33 Cfr. § 4.4 e successivi.

con l'installazione di Linux sulle postazioni, gli utenti interessati in questa fase sono stati suddivisi in gruppi di 8-12 persone; le “classi” hanno seguito corsi distinti in moduli da 3-4 ore durante il normale orario di lavoro e con argomenti di tipo principalmente “pratico”, come “Differenze terminologiche tra Windows e Linux” e “Differenze tra Microsoft Office e Openoffice”, strettamente orientate a problemi pratici nell'utilizzo dei nuovi sistemi. Durante la formazione, quindi, gli utenti hanno continuato ad utilizzare i sistemi con piattaforma Windows. Al fine di non rallentare la quotidiana produttività e consentire un approccio più “consapevole” ai nuovi sistemi, gli utenti hanno avuto modo di utilizzare le postazioni Linux solo dopo un ciclo di quattro giorni di lezione. Per tutto il periodo della formazione, il dipartimento IT ha reso disponibile una linea telefonica dedicata al supporto e all'assistenza per tutti gli utenti interessati dalla migrazione. Al fine, poi, di poter gestire al meglio le richieste di assistenza, il progetto ha previsto l'installazione di un software di *Trouble Ticket System*³⁴; la configurazione scelta per questo strumento è stata quella di permettere a tutti gli utenti di visualizzare le diverse richieste di assistenza, suddivise per area tematica, e le risposte dei tecnici. Utilizzando le informazioni acquisite, inoltre, è stata prevista una sezione delle domande poste più frequentemente (*Frequently Asked Questions, FAQ*) per offrire una prima forma di assistenza agli utenti in difficoltà. Al termine del processo di formazione di ogni classe, si seguivano le stesse fasi per i

34 Cfr. § 4.4.5.

restanti utenti e postazioni.

L'ostacolo più complesso da superare è stato l'approccio assunto da alcuni dipendenti sin dalla fase iniziale del progetto "Linux im Rathaus": in molti casi, infatti, è stata avvisata dal dipartimento IT una notevole riluttanza nei confronti di Linux e delle nuove applicazioni³⁵. Le prime difficoltà si sono manifestate chiaramente durante i primi corsi di formazione: alcuni partecipanti, infatti, hanno mostrato una notevole avversione nei confronti dei nuovi sistemi, accusandoli di "eccessiva complessità e di mancanza di intuitività". Notiamo come questo atteggiamento non fosse dipendente da effettive difficoltà nell'utilizzo delle postazioni: come illustrato in precedenza, infatti, il ciclo di lezioni di formazione era previsto in una fase precedente l'installazione di Linux sulle postazioni: gli utenti riluttanti, quindi, non avevano ancora utilizzato i nuovi sistemi.

Le conclusioni cui sono giunti i responsabili del progetto sono state che sarebbe stato possibile migliorare l'approccio alla migrazione da parte di alcuni utenti tramite un ciclo iniziale di incontri. Scopo di questi momenti sarebbe stato quello di mostrare direttamente, tramite videoproiettori e filmati esemplificativi, le presunte "difficoltà" di utilizzo dei nuovi sistemi e al fine di stimolare l'interesse dei partecipanti. Con questo tipo di incontri preliminari, il dipartimento IT avrebbe potuto mostrare rapidamente come gli ostacoli nell'utilizzo di Linux siano il più delle volte riconducibili alla

³⁵ In molti aspetti, l'atteggiamento mostrato da alcuni dipendenti dell'amministrazione comunale di Schwaebisch Hall si è simile a quanto incontrato in altre migrazioni. Abbiamo affrontato questo tema, denominandolo "Fattore Umano" nel § 4.5.1.

“diversità” di alcune operazioni piuttosto che alla loro effettiva complessità³⁶.

I responsabili del progetto, poi, si sono resi conto di un altro problema: gli utenti interessati dalla migrazione, più di duecento nel volgere di pochi mesi dalle fasi iniziali, non avevano modo di utilizzare Linux al di fuori dell'orario di lavoro. A differenza di quanto avveniva prima con i sistemi Windows, quindi, i dipendenti dell'amministrazione comunale di Schwaebisch Hall erano costretti ad utilizzare “a casa” un sistema operativo diverso da quanto usato “in ufficio”. In questo caso i provvedimenti furono presi immediatamente: furono distribuite copie di della distribuzione *Live Knoppix Linux Live CD*³⁷ e di Cd-Rom contenenti alcune applicazioni utilizzate in ufficio, come Openoffice e Mozilla Firefox, nella versione per Windows e per Linux. In questo modo, gli utenti avevano la possibilità di usare sui propri computer di casa i nuovi sistemi e prendere dimestichezza con Linux e le nuove applicazioni.

6.5 Conclusioni

36 Cfr. § 4.5.1.

37 Come illustrato in precedenza, i *Live CD* sono dei Cd-Rom contenenti il sistema operativo Linux e le applicazioni più importanti. A differenza dei normali supporti di installazione, questo tipo di sistemi possono essere avviati direttamente dal lettore Cd-Rom e non necessitano la scrittura sul disco fisso; in questo modo, l'utente può semplicemente inserire il Cd, avviarlo ed esplorare il funzionamento del nuovo sistema operativo senza correre il rischio di compiere operazioni potenzialmente dannose sul proprio computer. Il dipartimento IT dell'amministrazione comunale di Schwaebisch Hall ha offerto questo tipo di strumento al fine di permettere agli utenti di prendere dimestichezza con i nuovi sistemi, utilizzando i *Live CD* a casa propria. Cfr. § 4.5.1.

L'implementazione effettiva di Linux sulle postazioni client, all'interno del progetto "Linux im Rathaus", ha avuto inizio nell'estate del 2003. Il primo dipartimento interessato dalla migrazione, oltre naturalmente a quello IT, è stato l'ufficio del Sindaco con alcune postazioni correlate. Il processo poi ha interessato la biblioteca centrale, l'ufficio contabilità, l'ufficio anagrafe, la camera di commercio, il dipartimento dell'amministrazione centrale e gli altri uffici. Le sedi più distaccate, come ad esempio le postazioni all'interno degli asili comunali o le biblioteche comunali decentrate, sono state interessate dal processo solo nella sua fase finale. Per quanto riguarda le sedi di altre strutture collegate all'amministrazione comunale, è stato deciso di lasciare la scelta al direttore locale; tra queste, nell'autunno del 2004 hanno aderito al progetto "Linux im Rathaus" l'ufficio turistico e del marketing e altre sedi hanno intrapreso la prima fase della pianificazione preliminare. Ad oggi tutti i dipartimenti più importanti e le sedi distaccate direttamente dipendenti dall'amministrazione comunale hanno raggiunto la fase della migrazione completa, implementando Linux su tutte le postazioni desktop dei dipendenti³⁸.

Concludiamo l'esposizione di questa importante esperienza condotta dall'amministrazione comunale di Schwaebisch Hall con le parole del Ministro degli Interni tedesco Otto Schily, in occasione della conferenza del 2002 "*Open Source nella Pubblica Amministrazione*":

³⁸ Dati aggiornati al Marzo 2005.

“Aumentiamo la sicurezza informatica del nostro Paese, evitando la “monocultura”; riduciamo sensibilmente la nostra dipendenza da un singolo fornitore di software e risparmiamo sul costo delle licenze e sulle spese di manutenzione dei sistemi. Queste sono solo le premesse per l'importante sviluppo che il nostro Paese sta conoscendo nel settore dell'Information Technology”.

Conclusioni

Molto probabilmente gli utenti dei prodotti Apple, da tempo confinati in una ristrettissima nicchia di mercato, sorrideranno mentre osservano le ultime evoluzioni che sta avendo lo “scontro” tra Microsoft con i suoi prodotti da una parte e Linux dall'altra; sanno già che la minor diffusione di un software non implica la sua scarsa qualità. Una delle caratteristiche di Apple è da sempre stata quella di non preoccuparsi troppo di mantenere i propri prodotti all'interno di una nicchia: gli utenti ne sono consapevoli e ciò nonostante continuano a premiare l'azienda di Steve Jobs per le indubbe qualità mostrate dal sistema operativo MacOS e dagli applicativi ad esso dedicati.

Il fulcro è esattamente questo: i sistemi operativi, ed il software in generale, devono essere valutati in base alle caratteristiche e qualità mostrate. La scelta di un prodotto solo in virtù della sua diffusione non rappresenta una scelta consapevole, ma una mera accettazione dello *status quo*. Una decisione simile può essere condivisa solo qualora avvenga in assenza di valide alternative: se l'utilizzo di un tale prodotto comporta anche inefficienze, limitazioni e rischi in capo all'utente, difficilmente si potrà giustificare la decisione in maniera razionale.

Il mondo dell'Information Technology deve liberarsi dalla “*monocultura*” - utilizzando le parole del Ministro tedesco Otto Schily - dei prodotti

Microsoft nella quale da troppo tempo è caduto; l'alternativa a Windows esiste ed è rappresentata da Linux e dal software open source. Linux non è solo paragonabile al prodotto Microsoft, ma in molti aspetti è superiore ed in grado di apportare importanti benefici a chi deciderà di implementarlo.

La monocultura ha creato distorsioni tali da indurre molti soggetti a pensare semplicemente che non esistano “altri” sistemi operativi o che in ogni modo non siano confrontabili per semplicità e caratteristiche; anni di dominio assoluto del mercato, seppur per lungo tempo giustificato dalle qualità dei prodotti, hanno avuto inevitabili conseguenze sugli utenti e sulle loro abitudini.

La decisione in merito all'organizzazione di un network aziendale e dell'intero sistema informativo, però, non può limitarsi alla semplice accettazione della realtà: gli investimenti in queste risorse costituiscono sempre più un importante capitolo di spesa di ogni azienda. L' Information Technology è in continua evoluzione, è rapida e deve essere seguita attentamente: non è consentito rimanere indietro. Linux ha dimostrato a questo settore tutto il suo valore e le sue qualità: cominciando con il segmento dei server, ha rapidamente conquistato importanti quote di mercato ed un software open source, Apache web server, è il più diffuso strumento utilizzato come “motore” della rete Internet.

I vantaggi dell'installazione di Linux nei server rappresentano solo una minima parte di quelli conseguibili con l'implementazione sui desktop:

questo tipo di macchine, per definizione, costituiscono solo una piccola frazione dei computer presenti all'interno di un network aziendale. Il passo successivo, ed il più importante, è l'implementazione sulle singole postazioni.

La monocultura ha creato distorsioni tali da ritenere “normale”, o quanto meno inevitabile, sostenere importanti costi in termini di licenze d'uso per il software solo per rendere una postazione in grado di compiere normali attività di segreteria. Fino a qualche anno fa, però, non era presente una alternativa efficiente a Windows, in termini di costo e di caratteristiche, per i sistemi desktop: la maturità raggiunta da Linux e dai suoi applicativi rende finalmente possibile questa nuova sfida.

La sicurezza informatica, la carenza di bug nel codice, la rapidità nello sviluppo, la stabilità, il supporto, la flessibilità, la facilità di manutenzione, i risparmi per i costi di licenza e per altri aspetti, il *vendor tie-in*, l'indipendenza, la libertà di scelta. Nel nostro lavoro abbiamo argomentato quali siano le motivazioni che rendono Linux estremamente valido ed un'opportunità concreta per le aziende. La maggior parte di queste, però, presenterà già soluzioni basate su altri prodotti, molto probabilmente Microsoft; il nodo principale, quindi, non è semplicemente costituito dall'adozione di Linux e del software open source, ma la migrazione dei sistemi. Un'operazione complessa, ricca di imprevisti e da non sottovalutare, pena il pesante rischio di insuccesso.

L'implementazione di Linux non sarà limitata allo studio della sua fattibilità o convenienza, ma anche alla possibilità, per le aziende, di cambiare ed effettuare la migrazione nella maniera più efficiente ed indolore. Alla luce di queste ragioni, abbiamo mostrato nel nostro lavoro quali debbano essere le fasi di questo complesso ed importante processo: l'attenzione da dedicare all'analisi preliminare ed alla pianificazione della migrazione costituiscono solo la prima fase dell'implementazione di Linux in azienda. Grazie a queste operazioni sarà possibile rendersi effettivamente conto di quali possano essere i vantaggi conseguibili, in che misura, e quali potrebbero essere i rischi e le difficoltà da affrontare. Successivamente, la pianificazione permetterà di decidere quali debbano essere le strategie, le applicazioni e le tecnologie di migrazione cui fare ricorso per l'implementazione finale di Linux sulle postazioni.

In questa fase, così come in quella immediatamente successiva della migrazione, non dovrà mai essere sottovalutato il cosiddetto “fattore umano”; sarà necessario considerare e valutare attentamente quali potrebbero essere le conseguenze sugli utenti finali delle postazioni e quali le loro reazioni. Per far fronte ai rischi più comuni, abbiamo esposto quali potrebbero essere gli accorgimenti, tecnici e non, per superare le difficoltà legate alla preparazione ed alla formazione degli utenti, uniti alla particolare attenzione da dedicare alla configurazione delle postazioni. Strettamente legata a queste fasi, poi, la raccolta del *feedback* e il suo utilizzo al fine di

apportare eventuali variazioni al processo durante la sua stessa implementazione.

Siamo certi, infine, che l'esposizione e l'analisi dell'esperienza conosciuta dall'amministrazione comunale di Schwaebisch Hall per la migrazione completa dei sistemi, possa mostrare chiaramente quali possano essere le caratteristiche di un processo di questo tipo e, soprattutto, indicare quali siano i benefici che l'implementazione di Linux può apportare ad un network aziendale.

Concludiamo il nostro lavoro con l'auspicio che il mondo dell'Information Technology abbandoni la monocultura, che per definizione non dovrebbe appartenervi, e possa finalmente valutare, con coscienza e obiettività, lo scenario ormai maturo rappresentato da Linux e dal software open source.

Bibliografia

- AA.VV., *Approaching the Linux Desktop - A Novell Linux Migration Study*, Waltham, Novell Press, 2004.
- AA.VV., *Linux vs Windows: Total Cost of Ownership Comparison*, Victoria, Cybersource, 2004.
- AA.VV., *Open Sources: voci dalla rivoluzione open source*, Apogonline, 1999, disponibile su <http://www.apogonline.com/ebook/90016/ebook/html>
- Almond, C., Cannon, A., van Hoof, J., Mark, O., Patsch, T., Schwaller, T., Vaddadi, S., *Linux Client Migration Cookbook*, IBM Redbooks, Dicembre 2004, disponibile su <http://www.redbooks.ibm.com/redbooks/pdfs/sg246380.pdf>
- Bailey, M., Turner, V., Bozman, J., Waxman, J., *Linux Servers: What's the hype and What's the Reality?*, Framingham, IDC, 2000.
- Bärwolff, M. , Gehring, R. A. , Lutterbeck, B., *Open Source Jahrbuch 2005: Zwischen Softwareentwicklung und Gesellschaftsmodell*, Berlino, Lehmanns Media, 2005.
- Berra, M., Meo, A.R., *Informatica solidale: storia e prospettive del software libero*, Torino, Bollati Boringhieri, 2001.
- Biggs, M., *Open Source gets serious: a new window on Linux*, FCW.com, 2004, disponibile su <http://www.fcw.com/fcw/articles/2004/0308/feat-linux-03-08-04.asp>
- Bona, A., Disbrow, J.B., Butler, A., Bittman, T.J., *Hardware Improvements Drive Up Software Costs, Force Price Changes*, GartnerGroup, 2004.

- Christensen. C.M., *The Innovator's Dilemma: when new technologies cause great firms to fail (Management of Innovation and Change Series)*, Harvard, Harvard Business School Press, 1997.
- Coverity, “*Analysis of the Linux kernel*”, disponibile su www.coverity.com
- Covey, J., *A new Business Plan for Free Software*, Freshmeat, 22 Febbraio 2000.
- Downes, L., Mui, C., *Unleashing the Killer App: digital strategies for market dominance*, Harvard, Harvard Business School Press, 1998.
- F-Secure Computer Virus Database, disponibile su <http://www.f-secure.com>
- Gartner, “*Hardware Improvements Drive Up Software Costs, Force Price Changes*”, GartnerGroup, 15 Ottobre 2004, disponibile su www.gartner.com
- Gartner, “*Microsoft Stance on Multicore Licensing Will Help Customers*”, GartnerGroup, 20 Ottobre 2004, disponibile su www.gartner.com
- Gehring, R.A., Lutterbeck, G., *Open Source Jahrbuch 2004: Zwischen Softwareentwicklung und Gesellschaftsmodell*, Berlino, Lehmanns Media, 2004.
- Gould, J., *Will Enterprise Linux Stop Windows?*, Peerstone, 22 Ottobre 2004, disponibile su www.peerstone.com/pdfs/Peerstone_Ent_Linux_to_End_Win_Server_Growth.pdf
- Gustafson, P., Koff, W., *Open Source: Open for business*, El Segundo, CSC, 2004.
- Hardy, M., *Linux wants to earn your trust*, FCW.com, 2004, disponibile su <http://www.fcw.com/fcw/articles/2004/1018/tec-linux-10-18-04.asp>

- Hardy, M., *Linux weighs in*, FCW.com, 2004, disponibile su <http://www.fcw.com/fcw/articles/2004/0503/feat-linux-05-03-04.asp>
- Hochstetler, S., Arasaratnam, O., Castello Branco, R., *Linux: Why it should replace your NT Domains*, IBM Redbooks, Dicembre 2003, disponibile su <http://www.redbooks.ibm.com/redpapers/pdfs/redp3779.pdf>
- HoneyNet Project, *Know your Enemy: Trend Analysis*, HoneyNet Research Alliance, 17 dicembre 2004, disponibile su <http://www.honeynet.org/papers/trends/life-linux.pdf>
- Joch, A., *The Real Cost of Open Source*, FCW.com, 2004, disponibile su <http://www.fcw.com/fcw/articles/2004/1122/feat-open-11-22-04.asp>
- Kotadia, M., *Linux Total Cost of Ownership row continues*, Sidney, ZDNet, 2004, disponibile su <http://news.zdnet.co.uk/software/linuxunix/0,39020390,39180701,00.htm>
- Loftus, J., *The time is right for the open source in enterprise*, IT Consultant, 2004, disponibile su http://searchenterpriselinux.techtarget.com/qna/0,289202,sid39_gci1036077,0.html
- McMillan, R., *Analysis: The business case for desktop Linux*, IDG, 2004.
- Mitchell Smith, D., Simpson, R., Silver, M.A., Fiering, L., *Linux on desktop: the whole story*, GartnerGroup, 2003, disponibile su http://www.gartner.com/DisplayDocument?ref=g_search&id=406459
- Mitnick, K.D., *Time to Live on the Network*, San Francisco, Avantgarde, 2004.
- Mitnick, K.D., *L'arte dell'inganno*, Milano, Feltrinelli, 2003.

- Negroponte, N., *Essere digitali*, Milano, Sperling paperback, 1999.
- Novell, *Amsterdam Airport Schiphol*, Novell Linux Success Story, 2004, disponibile su www.novell.com/success/amsterdam_airport_schiphol.html
- Novell, *City of Bergen*, Novell Linux Success Story, 2004, disponibile su www.novell.com/success/city_of_bergen.html
- Novell, *City of Stockolm*, Novell Linux Success Story, 2004, disponibile su www.novell.com/success/city_of_stockholm.html
- Novell, *De Montfort University*, Novell Linux Success Story, 2004, disponibile su www.novell.com/success/demontfort.html
- Novell, *Duke University School of Engineering*, Novell Linux Success Story, 2004, disponibile su www.novell.com/success/duke.html
- Novell, *Harbor Federal Bank*, Novell Linux Success Story, 2004, disponibile su www.novell.com/success/harbor_federal_bank.html
- Novell, *Oregon Department of Transportation*, Novell Linux Success Story, 2004, disponibile su www.novell.com/success/odot.html
- Novell, *State of Nebraska*, Novell Linux Success Story, 2004, disponibile su www.novell.com/success/state_of_nebraska.html
- O'Reilly, T., Dyson, E., *Open Mind, Open Source*, O'Reilly Associates, 1999.
- Occorsio, Eugenio, *Affari & Finanza*, *La Repubblica*, Milano, 17 gennaio 2005
- Parck, A.L., Mein, J., *Microsoft Stance on Multicore Licensing Will Help Customers*, GartnerGroup, 2004, disponibile su http://www.gartner.com/DisplayDocument?ref=g_search&id=457015

- Quinlan, D., *The Past and the Future of Linux Standards*, Linux Journal, 1999.
- Raymond, E.S., *Colonizzare la noosfera*, Apogonline, 1998, disponibile su <http://www.apogonline.com/openpress/doc/homesteading.html>
- Raymond, E.S., *Il calderone magico*, Apogonline, 1998, disponibile su <http://www.apogonline.com/openpress/doc/calderone.html>
- Raymond, E.S., *La cattedrale e il bazaar*, Apogonline, 1998, disponibile su <http://www.apogonline.com/openpress/doc/cathedral.html>
- Raymond, E.S., Perens, B., *The Open Source Definition*, disponibile su www.opensource.org
- Shaw, R., *The TCO of Open Source*, CIO Today, 2005, disponibile su http://linuxtoday.com/it_management/2005010700126OSBZMR
- Slater, D., *Deciding Factors: Operation System*, CIO Magazine, 1 Febbraio 2000, disponibile su <http://www.cio.com/archive/020100/opsys.html>
- Stallman, R., *GNU General Public License (GPL)*, Free Software Foundation, disponibile su <http://www.gnu.org/licenses/gpl.txt>
- Stallman, R., *Philosophy of the GNU Project*, GNU Foundation, disponibile su <http://www.gnu.org/philosophy>
- Stallman, R.M., *Software libero, pensiero libero*, Viterbo, Nuovi Equilibri, 2001.
- Stoltz, M., *The Case for Government Promotion of Open Source Software*, NetAction White Paper, 1999, disponibile su <http://www.netaction.org/opensrc/oss-case-index.html>
- Terpstra, J.H., *Mission Critical Experiences with Linux*, Bynari, 2000,

disponibile su www.bynari.com/bcg/cases/mclinux.html

- Varian, H.R., Shapiro, C., *Linux Adoption in the Public Sector: An Economic Analysis*, Berkeley, Berkeley Press, 2003, disponibile su <http://www.sims.berkeley.edu/~hal/Papers/2004/linux-adoption-in-the-public-sector.pdf>
- Weiner, B., *Open Benchmark: Windows NT Server 4.0 and Linux*, Mindcraft, 1999, disponibile su <http://www.mindcraft.com/whitepapers/openbench1.html>
- Weiss, G., *The Gartner Group Server Operation System Forecast*, Gartner Group, 2000.
- Wheeler, D.A., *Why Open Source Software? Look at the Numbers!*, 2005, disponibile su http://www.dwheeler.com/oss_fs_why.html
- Wikipedia, *Timeline of notable computer viruses and worms*, disponibile su http://en.wikipedia.org/wiki/Timeline_of_notable_computer_viruses_and_worms
- Williams, S., *Codice Libero (Free as in Freedom)*, Apogonline, 2003, disponibile su <http://www.apogonline.com/ebook/90045/ebook/html>
- Winslow, M., *The practical manager's guide to Open Source*, Morrisville, Lulu Press, 2004.