

## Adaptive Internet Protocol

X, Java ed altro su tutti i client, oggi  
in Intranet, domani in Internet

di Leo Sorge

Dentro e fuori Internet esistono molti protocolli di trasmissione dati, ognuno con le sue caratteristiche, alcune vantaggiose, altre penalizzanti. I tradizionali protocolli del mondo Unix sono X in rete locale ed NFS in rete geografica, considerando il TCP/IP ormai un patrimonio di tutti. Internet ha lanciato l'HTTP e l'FTP, grazie ai quali il client e il server comunicano tra loro.

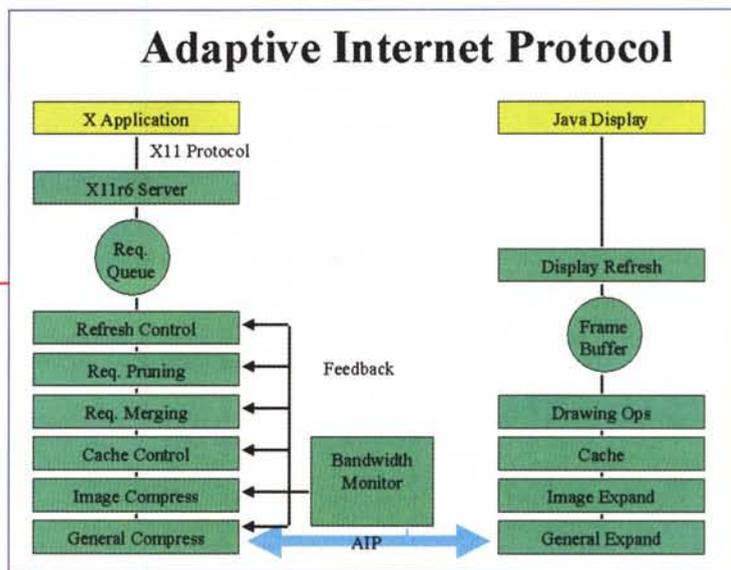
Tutti questi protocolli, progettati in ere più o meno remote, continuano ad evolvere. IP ha ormai pronta la versione 6 che ci affrancherà dal problema degli indirizzi mancanti, mentre NFS viene attaccato da più fronti tra i quali quello dei sostenitori del CIFS, Common Internet File Sharing.

Negli ultimi tempi però si è verificata una necessità di tipo nuovo. Dopo l'ubriacatura dell'informatica personale, nella quale i dati erano locali e le comunicazioni ridotte al minimo, si sta tornando ad una visione di tipo opposto: dati e programmi in un sito centrale, con il solo *data entry* in qualche modo spostato in locale. Una bella rete virtuale, con molte LAN connesse tra loro grazie a backbone terrestri o satellitari, rende attiva una Intranet sparsa un po' dappertutto. Eventualmente qualche servizio viene realizzato anche sulla stessa Internet.

Lo svilupparsi di questo nuovo paradigma, o meglio la necessità d'implementare il mainframe con l'hardware e il software messi in circolazione negli ultimi dieci anni, ha portato ad alcune conseguenze, riassumibili nelle due categorie software e hardware.

1) Dal punto di vista del software si è generata una nuova generazione di concetti server, assorbiti da Notes (che ha abbracciato Internet), NT (che sta abbracciando la multiutenza) ed Unix

Bellissima l'idea dietro l'AIP. Nato per risolvere il rapporto conflittuale tra il pesante protocollo X e le ridotte bande passanti disponibili in rete già locale, mostra una pipeline software che agisce in parte o in tutto a seconda della disponibilità della connessione. Futuri sviluppi prevedono altre applicazioni oltre all'attuale conversione X/Java, così come un tentativo di migrazione ad Internet.



(che s'è integrato con i mainframe) in modo diverso ma comunque attivo.

2) Dal punto di vista hardware c'è la necessità di connettersi a qualsiasi tipo di client.

Al riguardo la nuova tassonomia distingue quattro categorie:

a) *legacy client*, ovvero i vecchi PC (286/486), i terminali a carattere e quelli con X-Window;

b) *fat client*, ovvero i PC attuali con processori dal Pentium in su e configurazioni roboanti;

c) *thin client*, che comprendono i network computer (Javastation, Wyse, NCD...) e i netPC nella formulazione di Microsoft;

d) *anorexic* (ultra-thin) client, quali il Pilot di US Robotics o gli epigoni del Nokia 9000, in attesa degli HPC.

Tutti i protocolli client esistenti sono fatti per la categoria b) e si adattano in qualche modo alla categoria a), ma hanno problemi crescenti per le altre due. Non parliamo poi di cosa succede quando si passa su Internet. Per dirla con un'unica espressione, i protocolli attuali non scalano bene verso il basso.

Per risolvere questo problema sono stati presentati due nuovi protocolli, l'ICA di Citrix e l'AIP di SCO. Inizialmente SCO avrebbe dovuto sposare ICA, ma poi ha pensato bene di

sviluppare un suo progetto ottimizzato che lavorasse bene su X e su Java ma permettesse una semplice estensione a nuove necessità. AIP, Adaptive Internet Protocol, è una delle tecnologie alla base di Tarantella, il server Intranet di SCO. E' di questo protocollo che parliamo nel seguito.

### AIP, Adaptive Internet Protocol

Nato come AAP, Advanced Adaptive Protocol, ha in seguito cambiato nome rivolgendosi ad Internet. Non si hanno però notizie né d'una richiesta di standardizzazione né della formazione d'un SIG (Special Interest Group), per cui dovrebbe essere a tutti gli effetti un protocollo proprietario. Il progetto è così limpido che è possibile riassumerne le caratteristiche salienti in un breve elenco:

- indipendenza dal client;
- indipendenza dalla connessione;
- semplicità d'implementazione di nuovi clienti e nuove connessioni;
- scoperta automatica del client;
- ottimizzazione delle prestazioni attraverso il controllo della linea.

È alle funzionalità dell'ultima riga che deve il nome, in quanto analizzando

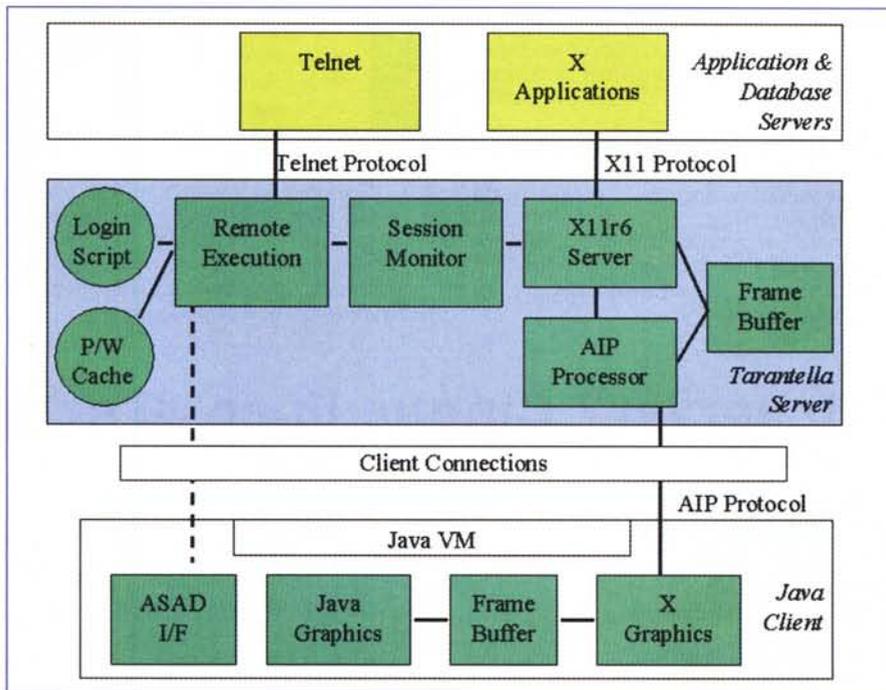
costantemente la disponibilità di banda il protocollo inizia a trasmettere componenti appropriate, evitando di svolgere lavori complessi in momenti nei quali la connessione è labile. Ciò presuppone anche una suddivisione dei compiti in più fasi indirizzabili separatamente, che come vedremo è la grande novità di questa tecnologia.

Nell'architettura a tre strati imposta da Tarantella alle reti aziendali, AIP viene usato per la comunicazione con i client, locali o su rete geografica. Il protocollo è reso funzionale da un processore che almeno per ora è implementato in software.

Ecco il modello logico di riferimento per la coppia X/Java, ritenuta fondamentale nella prima implementazione. L'applicazione X, ovviamente connessa al pesante motore X11r6, viene analizzata tramite una pipeline software attualmente composta di sei stadi principali pensati per il compito specifico. Ad esempio la compressione viene svolta in due passaggi, uno per le immagini e l'altro per ciò che resta. Questa implementazione è funzionale al vero cuore della soluzione, il *bandwidth monitor*, che controlla la linea di trasmissione per conoscerne la capacità del momento. Confrontando la situazione misurata con un modello interno di trasmissione ottimizzata, il monitor stabilisce quale fase mandare avanti e quale componente trasmettere.

La figura successiva mostra AIP all'interno del modello a tre livelli.

*Nel modello a tre livelli di Tarantella, AIP è il mezzo unico a bassa velocità di comunicazione con i client, quali essi siano.*



li di Tarantella, sempre nello specifico d'una applicazione X.

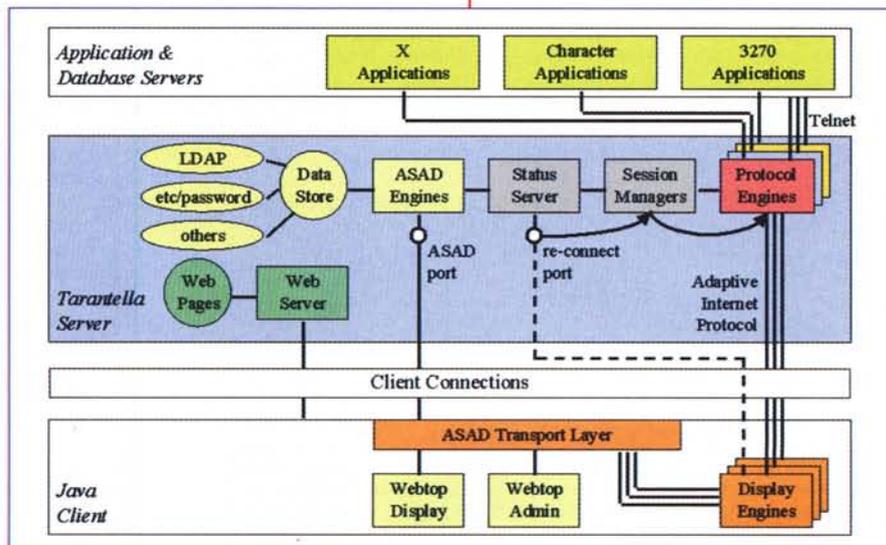
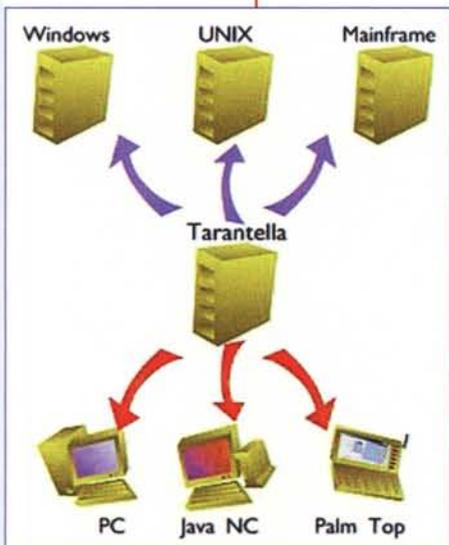
Come già accennato, l'Intranet server di SCO impone un modello a tre livelli, interponendo Tarantella tra gli altri server e i client. Questa scelta, che aggiunge un livello al tradizionale modello client/server, è stata criticata sia dalla concorrenza che dagli sviluppatori, essendo credo comune che aggiungere un intermediario complicherebbe e rallenterebbe. Al riguardo la posizione di SCO è che questa struttura, oltre a permettere alcune scelte (ad esempio per la sicurezza) non possibili con due soli livelli, non è penalizzata nelle pre-

stazioni per la struttura di AIP e per la differente velocità tra il legame con i server superiori rispetto a quelli con i client.

SCO ha implementato in laboratorio una soluzione particolare, scorpendo l'X-server dal resto di Tarantella e mettendo le due parti come server applica-

*Una sessione di riferimento per il protocollo X, la vera eredità pesante (per la rete) di Unix.*

*Attendiamo con ansia il piano rivolto alle terze parti desiderose d'implementare loro versioni della coppia PE/DE appoggiandosi su AIP. Solo il futuro ci dirà qualcosa del confronto tra questo protocollo e l'ICA di Citrix.*



tivi, quindi eliminando il livello intermedio. Le prestazioni non sono cresciute, ma si è sentita la mancanza delle funzionalità rese possibili da un livello intermedio.

Dal punto di vista degli sviluppatori AIP si compone della coppia protocol engine/display engine: il primo resta nel server, il secondo va sul client. SCO sta studiando un ricco programma tecnico per diffondere il modello e aumentare la disponibilità di protocol/display.

MS

## o LINUX

g  
n  
a  
L'

(Roma, 3 settembre) P'ppiniello s'ha 'mpazzito... ho pensato leggendo quello che mi proponeva come pezzullo e che è molto di più. Però è intrigante, io sto per installare OpenLinux (in questo caso "sto" identifica un periodo iniziato tre mesi fa e mai finito) e sto per recensire (credo sulle frequenze di questa rubrica) un simpatico libro edito da O'Really che parla proprio di come scrivere Web client e server. Tutto ciò ha fatto pendere l'ago della bilancia verso la pubblicazione d'un qualcosa astruso ed inutile ma intrigante ed illuminante. Sogghigno sadicamente nell'andare ad amputarne brutalmente parti scelte a casaccio.

PS (Roma, 22 settembre) Nelle ultime due settimane Linux è balzato agli onori delle mie cronache, con svariate e-mail di protesta e non, in parte pubblicate nella rubrica della posta (a proposito, mi scuso con Andrea Balestrero, Riccardo Ottonello e Alessandro Conti, ai quali non credo di aver risposto), poi con la venuta in Italia dei rappresentanti di Caldera, ricchi di novità. Neanche quando intervistai Linus Torvalds ebbi tanto da fare su questo argomento!

di Leo Sorge

## Web server? Cinque linee di script!

di Giuseppe Zanetti

Lungi da me trattare in questa rubrica argomenti più consoni al newsgroup comp.os.linux.advocacy. Per una volta vorrei però essere più tecnico, sfruttando questo spazio per spiegare perché alle volte possa risultare conveniente la scrittura di un programma in Linux piuttosto che in un altro sistema operativo.

Infatti per offrire determinati servizi, anche in una rete formata da macchine monomarca, spesso può convenire l'uso di sistemi operativi che per quella rete non sono standard. Linux infatti, oltre ad una notevole robustezza, dispone di alcune caratteristiche peculiari, come la possibilità di programmare applicazioni molto velocemente, usando tool standard e modulari e grazie alla possibilità di riuso di codice sorgente soggetto alla licenza GPL e quindi di uso libero (se non commerciale, ndr). Grazie al TCP/IP, i programmi realizzati saranno comunque facilmente integrabili con il resto della rete.

Le possibilità di interfaccia con le macchine client sono notevoli: dall'accesso mediante emulazione di terminale alla mappatura del servizio su una porta TCP/IP e accesso ad esso mediante programmi scritti allo scopo o mediante un protocollo standard, ad esempio TCP/IP. Per alcuni tipi di elaborazione si può pubblicare il risultato direttamente su un filesystem condiviso (e Linux supporta benissimo la condivisione di file da e verso Lan Manager) oppure l'accesso da parte di una rete macchine di macchine Windows ad una applicazione di FAX sotto Linux condividendo il fax virtuale come se fosse una stampante Lan Manager.

La realizzazione di un programma che offra un semplice servizio accessibile mediante TCP/IP è in Linux molto semplice da realizzare e non richiede l'acquisto di costose librerie. Nei casi più semplici non è neppure necessario programmare in ambiente TCP/IP, in quanto è possibile scrivere il proprio programma in modo che funzioni prendendo le informazioni da standard input e scrivendo i risultati nello standard output, come un qualunque programma scritto per lavorare da terminale. Ci penserà poi il "superserver" **inetd** a fare da interfaccia fra il nostro server e la rete.

Un semplice programma che funzioni da server WWW può essere perciò realizzato mediante pochissime linee di shell script:

```
#!/bin/sh

BASE=/usr/httpd/htdocs

read COMMAND PAGE HTTP
cat $BASE$PAGE

exit 0
```

La prima riga indica che il programma deve essere eseguito dalla shell standard di Linux /bin/sh, mentre quelle seguenti accettano la richiesta passata dal client, nella forma "GET /path/doc.ext

HTTP/1.0" e stampano la pagina corrispondente. Ovviamente un server WWW vero dovrebbe fare altre cose, prima fra tutte spedire un header contenente il MIME type corretto per il documento richiesto... e, magari, anche fare qualche controllo per la sicurezza. Affinché il programma funzioni è sufficiente aggiungere in /etc/services l'associazione del nuovo servizio ad una porta TCP libera, ad esempio la 9999:

```
myhttpd 9999/tcp
```

In /etc/inetd.conf vanno inserite invece le informazioni riguardanti la modalità di esecuzione del server, l'utente di cui ereditare i permessi ed il path corrispondente:

```
myhttpd stream tcp nowait daemon /usr/local/bin/myhttpd
```

A questo punto si può mandare ad **inetd** un segnale che gli faccia rileggere i file di configurazione modificati:

```
killall -HUP inetd
```

Il nuovo servizio sarà disponibile all'URL <http://mi.ghost:9999/path/doc.ext> e potrà essere acceduto, mediante metodo standard, da client di qualunque tipo.

L'uso di TCP/IP ed altre metodologie Intranet permettono una coesistenza trasparente fra la macchina Linux e le altre macchine della rete, al punto che spesso è comodo che la realizzazione di tali servizi venga offerta nella forma di *black box*, ovvero di un sistema preconfezionato da collegare alla rete e da dimenticare: l'affidabilità di Linux ed un contratto di supporto col fornitore possono comunque fornire una soluzione ottimale, ad un costo concorrenziale e, soprattutto, realizzata con metodi standard, collaudati ed ampiamente documentati.

### Un altro esempio

A proposito di soluzioni arzigogolate, in Linux per collegare un programma in modo che elabori la posta a mano a mano che essa arriva ad un utente è sufficiente scrivere, nel linguaggio che ci è più congeniale, un programma che legga la mail ricevuta da standard input ed inserir il suo path nel file .forward nella home directory dell'utente prescelto, ad esempio /home/pippo/.forward:

```
"/usr/local/bin/postino"
```

In questo modo, per chi ha un po' di pratica di UNIX, diventa abbastanza semplice, ad esempio, sviluppare una interfaccia fra e-mail e Web o archiviare opportunamente i messaggi che arrivano ad una mailing list, oppure fare in modo che il computer faccia partire un processo, ad esempio comporre il numero di un cercapersone o spedire un SMS ad un cellulare, ogni qual volta riceve una data mail. Il fatto che ad ogni utente del sistema siano associate determinate permission fa sì che il programmatore non debba perder tempo a scrivere del codice che se ne occupi

# FINALMENTE SPIEGATA LA FOTOGRAFIA DIGITALE

L'evoluzione digitale della fotografia fornirà ai fotografi professionisti e agli amatori appassionati di computer molte nuove opportunità.

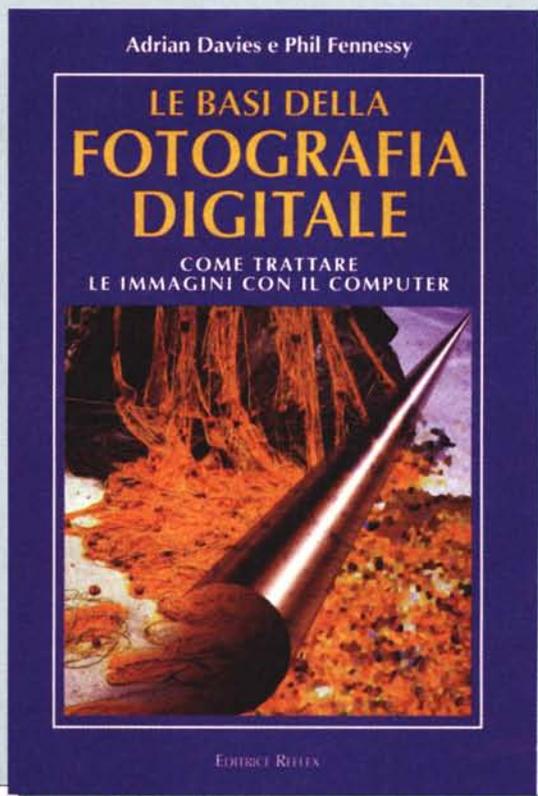
Questo libro spiega la tecnica degli strumenti (computer, scanner, stampa, trasmissione, ecc.) soffermandosi sul trattamento delle immagini. Ampio spazio è dedicato all'utilizzo degli strumenti di Photoshop ed al modo di avvicinarsi a questo eccellente programma di elaborazione dell'immagine.

Scritto da fotografi per i fotografi con un linguaggio semplice ed esauriente allo stesso tempo, il libro descrive con precisione e chiarezza tutto ciò che è necessario conoscere per orientarsi nel mondo della fotografia digitale. L'ampio glossario assicura che il significato di ogni termine sia perfettamente chiaro al lettore. Dopo, tutti i vostri dubbi sul fenomeno digitale saranno cancellati.

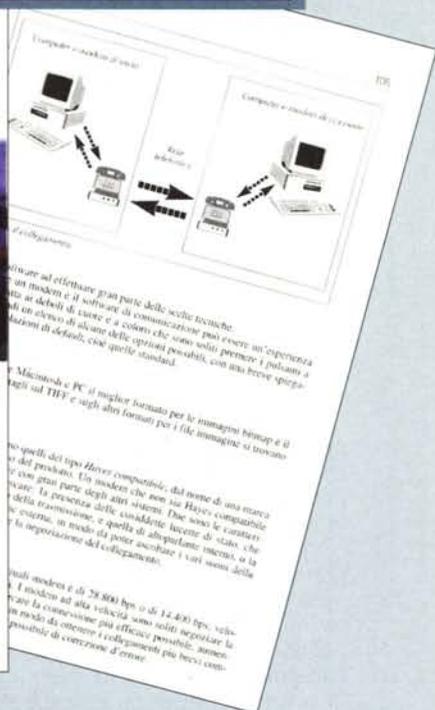
Dalla Editrice Reflex

## LE BASI DELLA FOTOGRAFIA DIGITALE

L. 36.000, 136 pagine, 15x21cm.



Alcune tecniche spiegate nel libro.



**IN VENDITA NEI MIGLIORI NEGOZI DI FOTOGRAFIA ED IN LIBRERIA.**

POTETE RICHIEDERE IL VOLUME DIRETTAMENTE ALLA EDITRICE REFLEX. PAGAMENTO CON ASSEGNO BANCARIO, CARTA DI CREDITO (AMERICAN EXPRESS o CARTASI) OPPURE VERSANDO L'IMPORTO SUL CCP N. 82707001 INTESTATO A:

EDITRICE REFLEX, VIA DI VILLA SEVERINI 54, 00191 ROMA

TEL. 06-36308595 - 36301756 FAX 06-3295648