DigItalia

Numero 2 - **2007**

Rivista del digitale nei beni culturali

ICCU-ROMA

Interpretare le statistiche Web

Andrea Giuliano
ICCU

Espressioni comuni come "sono collegato al sito tal dei tali", perfettamente accettabili e chiare nel contesto di tutti i giorni, in un contesto come quello delle statistiche Web possono creare più di un equivoco. In particolare, quando sto leggendo una pagina visualizzata dal mio browser, non sono affatto "collegato al sito". Quando accedo col browser ad una pagina Web di un sito, il browser inoltra al server Web un'opportuna richiesta, e il server risponde inviando la pagina in questione, oppure segnalando eventuali situazioni di errore. Il protocollo HTTP non stabilisce delle sessioni, ma apre una connessione per ogni file da trasferire, e la chiude appena terminato il trasferimento (a differenza del protocollo FTP che, nel corso di una sessione, consente di trasferire un numero qualsiasi di file e fare anche altre operazioni).
Per ciascuna richiesta che riceve, un server Web registra molte informazioni in un apposito file di log. Esistono numerosi programmi, sia commerciali che gratuiti, che analizzano i file di log ricavandone una vasta serie di dati statistici. Spesso i responsabili amministrativi si aspettano di poter estrapolare da questi dati informazioni preziose circa l'uso e il gradimento dei loro siti Web. Ma, in realtà, l'interpretazione di questi dati risulta attendibile solo alla luce di considerazioni molto restrittive.
Ad esempio, un'importante informazione registrata normalmente da un server Web è l'indirizzo IP del client, cioè del sistema che, attraverso il protocollo HTTP, ha richiesto al server stesso un qualche file. Verrebbe spontaneo pensare a questo indirizzo come a un visitatore: in fondo, gli indirizzi IP sono notoriamente univoci, per cui due persone che accedono al sito non possono avere lo stesso indirizzo IP. In effetti, quasi tutti gli analizzatori di log Web definiscono visitatore un indirizzo IP univoco, e riportano tabelle in cui sono conteggiati, fra l'altro, gli indirizzi IP distinti registrati nei log in un certo periodo di tempo. Da queste tabelle si può pensare di ricavare un'idea approssimativa di quante persone accedono al sistema nell'unità di tempo. Sarebbe molto bello se fosse così, ma da quando esistono i firewall questo quadro idilliaco è del tutto irrealistico. Perché, in questo caso, il server può registrare nel file di log solo l'indirizzo del firewall, senza essere in grado di vedere tutti gli altri indirizzi IP attivi dietro il firewall stesso.
Questo è uno dei tanti problemi che rendono il tema delle statistiche Web un argomento piuttosto complesso.

Introduzione

Molti sistemi informativi moderni sono accessibili via Web, e questo vale anche nell'ambito dei beni culturali. Gli accessi via Web sono registrati secondo modalità abbastanza standard, per cui da tempo sono nati applicativi che ana-

lizzano i dati relativi agli accessi e presentano quindi i risultati della loro analisi, le cosiddette "statistiche Web".
Esse costituiscono certamente un utile strumento per valutare il gradimento di un sito Web e il modo con cui esso è utilizzato. Tuttavia è molto facile interpretarle in modo errato, soprattutto se si ignorano i pochi, semplici ma ineludibili principi che sono alla base del funzionamento stesso del Web.
In questo lavoro si cerca di fornire, sia pure in forma approssimativa, alcuni di questi principi, al fine di aiutare i responsabili dei sistemi a valutare correttamente le statistiche Web.

In principio era il log

Al tempo in cui le navi inglesi stavano conquistando il predominio su tutti i mari, sia in termini militari che commerciali, rilevare periodicamente la velocità di una nave era estremamente importante, soprattutto nelle traversate oceaniche.
Prima che fossero introdotti strumenti più sofisticati, tale rilevamento era effettuato con uno strumento molto semplice: una cordicella lunga e leggera sulla quale erano praticati dei nodi a intervalli regolari e alla cui estremità era fissato saldamente un tronchetto di legno. Un marinaio gettava in mare il tronchetto e lasciava scorrere la cordicella fra le mani contando i nodi per un fissato periodo di tempo. Il numero di nodi contati veniva poi trascritto da un ufficiale in un apposito registro.
In inglese, un tronchetto come quello utilizzato in questo tipo di rilevamento prende il nome di *log*. Data l'importanza e la frequenza con cui veniva utilizzato per misurare la velocità delle navi, non trascorse molto tempo prima che il significato del termine si estendesse al registro stesso in cui venivano trascritte le misurazioni. Più tardi, al nascere dell'aviazione civile, esso si estese ulteriormente e ancora oggi si parla di *flight log*.
Oggi il significato di *log* è esteso anche all'informatica, e indica i file in cui certi applicativi registrano informazioni sulla loro attività, a scopi diagnostici, ma frequentemente anche a scopi statistici. È proprio questo il caso dei *log* dei siti Web.

I log di un server Web

Un sito Web, in parole molto povere, è una raccolta più o meno vasta e articolata di pagine Web, particolari file di testo scritti secondo le specifiche HTML (Hyper Text Markup Language) proposte dal W3C (World Wide Web Consortium). Queste specifiche consentono di creare e presentare testi arricchiti da tabelle, elementi grafici, sottolineature, carattere corsivo o grassetto, e moltissimo altro. Inoltre, le pagine possono essere collegate fra loro da collegamenti ipertestuali, o *link*, che consentono di superare la struttura sequenziale caratteristica del testo tradizionale per arrivare al cosiddetto *ipertesto*, le cui pagine non possono essere ordinate, ma costituiscono un albero o, più in generale, un grafo orientato.
Per la precisione, bisogna dire che un sito Web può contenere e rendere accessi-

bili file di qualsiasi tipo, anche se le pagine Web sono il tipo di file più comune gestito dai siti Web. Inoltre, come si vedrà nel seguito, consente anche di presentare contenuti dinamici, cioè dipendenti da informazioni fornite dall'utente, e non soltanto file *statici*.
Il server Web è quel particolare software che, secondo le regole specificate dal protocollo HTTP (Hyper Text Transfer Protocol, RFC 2616), trasmette su richiesta le pagine Web di un sito. Un *client* Web, invece, è il software usato da un utente di Internet per ottenere pagine dai siti che costituiscono il World Wide Web. Esempi molto diffusi sono Mozilla Firefox, Opera, Safari (su computer Apple) e Internet Explorer. Esistono molti altri *client* normalmente non utilizzati dagli utenti di Internet, ma comunque utilissimi per applicazioni particolari. Ad esempio, i motori di ricerca Web utilizzano *client* sviluppati allo specifico scopo di visitare e indicizzare automaticamente l'intero Web, senza intervento umano.
Vediamo ora cosa avviene quando si visita un sito, e quali informazioni vengono registrate nei log.
Quando l'utente accede col suo browser a una pagina Web di un sito, il browser inoltra al server Web che gestisce quel sito un'opportuna richiesta, e il server gli risponde inviando la pagina in questione, oppure segnalando eventuali situazioni di errore. Inviando al *client* il file richiesto, il server è nuovamente pronto a esaudire ulteriori richieste, da quel *client* come da qualsiasi altro.
Per ciascuna richiesta che riceve, un server Web registra molte informazioni in un apposito file di log. In un caso molto elementare, un tipico estratto dai *log* di un server Web si presenta così:

```
192.168.21.67 - - [13/Jun/2007:09:00:19 +0200] "GET /index.html HTTP/1.1" 200
44 "-" "Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.1.3) Gecko/20070310
Epiphany/2.14 (Debian-2.0.0.3-1)"
192.168.21.67 - - [13/Jun/2007:09:00:44 +0200] "GET /index.html HTTP/1.1" 200
44 "-" "Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.1.3) Gecko/20070310
Iceweasel/2.0.0.3 (Debian-2.0.0.3-1)"
192.168.21.67 - - [14/Jun/2007:18:39:53 +0200] "GET /" 200 44 "-" "-"
192.168.21.67 - - [14/Jun/2007:18:52:16 +0200] "GET /welcome.html HTTP/1.0"
200 109 "-" "Wget/1.10.2"
```

Le righe possono essere di varia lunghezza, e nell'esempio sopra sono state spezzate perché troppo lunghe. La separazione fra una riga e l'altra è stata riprodotta tramite una maggiore interlinea.
Chiarito questo, si dovrebbe notare facilmente che tutte queste righe hanno la stessa struttura. Per capire cosa rappresentano le varie parti di una riga di *log* conviene partire dall'evento che provoca la registrazione di una riga, e cioè dalla richiesta di trasmissione di una pagina Web. Le varie parti della riga saranno eviden-

ziate volta per volta. Con i punti di sospensione si ometteranno le parti meno interessanti, per non appesantire troppo la lettura.
Supponiamo ad esempio che un utente, lavorando su una postazione con indirizzo IP 192.168.21.67, voglia accedere a una particolare pagina Web del sito:

http://il.mio.sito/

Questo accesso corrisponderà a una specifica riga nel file di *log* del server Web che gestisce quel sito. In particolare, l'indirizzo IP della postazione dell'utente sarà registrato proprio all'inizio della riga:

192.168.21.67 - - [13/Jun/2007:09:00:19 +0200] ...

La data e l'ora in cui l'utente ha cercato di accedere alla pagina sono riportate poco dopo:

192.168.21.67 - - **[13/Jun/2007:09:00:19 +0200]** "GET ...

La specifica pagina richiesta dall'utente è indicata invece nella parte tra virgolette che segue:

... 09:00:19 +0200] "GET **/index.html** HTTP/1.1" 200 44 ...

In questa parte è anche indicata la versione del protocollo HTTP che il browser ha scelto di usare per parlare con il server. La versione 1.1 è quella universalmente usata da tutti i siti Web del mondo da diversi anni a questa parte:

... 08:57:53 +0200] "GET /index.html **HTTP/1.1**" 200 44 ...

Può succedere che il server non possa fornire questa pagina all'utente, magari perché l'utente ha scritto male l'indirizzo della pagina, oppure perché l'ha chiesta al sito sbagliato. Non essendo quindi scontato che la richiesta venga soddisfatta, è importante registrare l'esito di una richiesta, positivo o meno. Un esito positivo è indicato nella riga di *log* dal codice 200, seguito dalle dimensioni in byte del file trasferito:

... 08:57:53 +0200] "GET /index.html HTTP/1.1" **200 44** ...

mentre un esito negativo sarebbe indicato nella stessa riga di *log* dal codice 404. Ad esempio, se l'utente ha richiesto per errore la pagina http://il.mio.sito/ondex.html, la riga si presenterebbe così:

... 08:57:53 +0200] "GET /**ondex**.html HTTP/1.1" **404** 288...

Infine, il server Web registra nel *log* svariate informazioni relative al browser, informazioni fornitegli dal browser stesso. Ad esempio:

... 200 44 "-" "Mozilla/5.0 (X11; U; **Linux** i686; en-US; rv:1.8.0.11) Gecko/20070324 (Debian-1.8.0.11-2) **Epiphany/2.14**"

Oppure

... 200 109 "-" "**Wget/1.10.2**"

In entrambi i casi si possono riconoscere il nome e la versione del browser usato dall'utente (Epiphany/2.14 e Wget/1.10.2), mentre solo il primo browser si è curato di informare il server anche circa il sistema operativo installato sulla postazione dell'utente (Linux). In teoria, non solo il *client* non è obbligato a fornire questo genere di informazioni, ma è anche libero di fornirle come più gli piace. Ad esempio, in un *log* possono apparire anche righe come la seguente:

... "HEAD / HTTP/1.1" 200 - **"-" "Plexas was here!"**

che è stata registrata in seguito a una richiesta effettuata con il comando *telnet*, presente in tutti i sistemi operativi, ma che decisamente non si può considerare un *client* Web *user friendly*.

Analisi e interpretazione dei log

Esistono numerosi programmi, sia commerciali che gratuiti, che analizzano *log* come quello appena visto ricavandone una serie di informazioni. Spesso tali informazioni sono prese molto sul serio dai responsabili amministrativi e dai dirigenti, i quali pensano di poterne ricavare facilmente indicazioni circa l'uso e il gradimento di un sistema informativo acceduto tramite un sito Web. Ma quante di queste informazioni sono realmente attendibili? E come vanno interpretate?
Vedremo ora alcune di queste informazioni, spiegando come vengono ricavate dai log e valutandone il significato e l'attendibilità.

Accessi, non sessioni

Le poche righe di *log* mostrate e commentate poco sopra hanno in comune una caratteristica fondamentale: ognuna si riferisce alla richiesta di un file da parte del *client* e alla risposta del server a tale richiesta. Se il server dispone del file richiesto, lo invia al browser che lo presenta all'utente, altrimenti trasmette al browser una segnalazione di errore.

Conclusa questa transazione, il browser e il server non sono più in contatto: l'utente potrebbe staccare il modem o il cavo LAN mentre legge la pagina richiesta, e ricollegarlo quando deve visitarne un'altra. Infatti il protocollo HTTP è orientato alla connessione, ma non alla sessione.
Per meglio chiarire il concetto di sessione, si pensi al protocollo FTP (File Transfer Protocol), che invece è orientato alla sessione: l'utente accede al server FTP, con o senza autenticazione, e finché non esprime chiaramente la volontà di distaccarsi da esso, può scaricare o caricare quanti file vuole, o eseguire vari comandi, sempre all'interno di una stessa sessione di lavoro. Altri protocolli orientati alla sessione sono POP (Post Office Protocol), usato per scaricare la posta elettronica da un server, IMAP (Internet Mail Access Protocol), usato per consultare la posta da remoto, e SSH (Secure Shell), usato per lavorare in remoto su un sistema attraverso una connessione cifrata.
HTTP, invece, non stabilisce delle sessioni: apre una connessione per ogni file da trasferire, e la chiude appena terminato il trasferimento (a differenza di FTP che, nel corso di una sessione, consente di trasferire un numero qualsiasi di file e fare anche altre operazioni). Per migliorare le prestazioni, HTTP 1.1 consente di trasferire molti file con una sola connessione, ma questo non trasforma quella connessione in una sessione, solo la rende più efficiente. Questo è coerente con gli scopi per cui è stato introdotto il protocollo HTTP, cioè inviare file su richiesta: a ogni richiesta corrisponde una connessione che si chiude non appena il file richiesto, oppure un messaggio di errore, è stato inviato al *client* che ha avanzato quella richiesta.
Pertanto, espressioni comuni come "sono collegato al sito tal dei tali", perfettamente accettabili e chiare nel contesto di tutti i giorni, in un contesto come quello delle statistiche Web possono creare equivoci. In particolare, quando sto leggendo una pagina che è stata interamente visualizzata dal mio browser, in tutte le sue parti, non sono affatto "collegato al sito". In quel momento, il server del sito Web e il mio browser si ignorano completamente, e non esiste alcuna connessione fra il browser e il server, anche se ne sono esistite probabilmente parecchie nei pochi secondi necessari alla presentazione della pagina sul mio browser.
Concludendo, si può dunque dire che ogni richiesta, indipendentemente dal suo esito, costituisce un accesso al sito, e che tali accessi, salvo situazioni particolari che sono al di là degli scopi di questo lavoro, non sono organizzati a formare delle sessioni. Anche se per alcuni è difficile accettarlo, il fatto è che normalmente non si può sapere cosa ha fatto un utente durante una sessione, semplicemente perché non esiste alcuna sessione.
In più, ci sono delle complicazioni.

Pagine e non

Una pagina Web, come si presenta all'utente, non è costituita da un singolo file. Quasi sempre la rappresentazione grafica che si ha davanti è stata composta dal

browser a partire da un singolo file che ne richiama, tramite appunto i *link* ipertestuali, parecchi altri. Si pensi ad esempio al sito http://opac.sbn.it. La pagina per la ricerca base è innanzitutto composta da quattro sottopagine (*frame*), corrispondenti a quattro file che, insieme al file principale che richiama questi, portano già a cinque gli accessi che saranno registrati nei log. Vanno poi considerati i pallini rossi e blu che sono file come tutti gli altri: il browser deve scaricarli dal sito per poterli mostrare all'utente al posto giusto. Ci sono poi diversi bottoni, e tenendo conto di tutti questi elementi, ci si avvicina a una trentina di accessi solo per poter vedere la maschera di ricerca base.
La maggior parte dei prodotti di analisi dei *log* è abbastanza intelligente da conteggiare separatamente i file HTML e i numerosi file ausiliari che contribuiscono sì al *rendering* grafico di una pagina, ma che non ha senso considerare come accessi significativi: le migliaia di richieste del file *pallino_blu.gif* non interessano, anche se sono accessi a tutti gli effetti. Interessa invece sapere quante volte è stato richiesto il file *ricerca_base.html*, perché questi accessi indicano una precisa volontà dell'utente, mentre l'accesso al pallino blu, in un certo senso, non è volontario, ma necessario alla presentazione della pagina.
Normalmente, i *log analyzer* chiamano appunto *pagine* i file HTML e alcuni altri tipi, riportando separatamente gli accessi a questo tipo di file. Le immagini e altri file meno significativi vengono trattati a parte.

Cache

La prima versione del protocollo HTTP prevedeva che il browser scaricasse una pagina o comunque dei file incondizionatamente: se una pagina era la stessa del giorno prima, veniva scaricata comunque. Questo costituiva uno spreco di banda, tenuto conto del fatto che molti siti, ancora alla metà degli anni '90 del XX secolo, erano costituiti da pagine fisse, modificate piuttosto di rado.
La versione 1.1 di HTTP introdusse il meccanismo di *caching*: in pratica, il browser conserva sul sistema locale dell'utente i file scaricati di recente, e può in seguito limitarsi a chiedere al server se una certa pagina è stata aggiornata rispetto alla versione che conserva in locale (nella memoria detta appunto *cache*). Se il server risponde di no, non vale la pena di scaricare nuovamente la pagina, tanto vale presentare all'utente la stessa pagina del giorno prima, risparmiando tempo e banda.
Di regola, anche le richieste di questo tipo sono registrate da un server Web, che le marca tramite un codice particolare (304, Not Modified) che indica che la pagina richiesta in realtà non è stata trasmessa, ma che comunque è stata richiesta. Ad esempio, il seguente ritaglio di *log* si riferisce a due richieste identiche, effettuate in meno di un minuto:

... [17/Jul/2007:19:24:56 +0200] "GET /index.html HTTP/1.1" 200 44 "-" ...
... [17/Jul/2007:19:25:46 +0200] "GET /index.html HTTP/1.1" 304 - "-" ...

Nella prima richiesta, il browser non trova nella sua *cache* la pagina richiesta, e quindi questa viene effettivamente trasferita (codice 200 e lunghezza 44 byte). Nella seconda riga, il browser e il server concordano che non è necessario procedere al trasferimento (codice 304 e lunghezza indefinita), perché la pagina è nella *cache* del browser ed è aggiornata quanto quella presente sul sito.
Questo è un comportamento corretto, soprattutto al fine di analizzare i log, perché viene registrata comunque l'intenzione di qualcuno che in effetti ha visto la pagina, anche se non sa di aver visto una sua copia locale.
La *cache* può però costituire una complicazione perché il server può essere anche configurato in modo da non registrare richieste con codice 304. In questo modo l'analisi dei *log* potrebbe essere seriamente falsata: infatti molte pagine, specie sui siti per l'accesso a banche dati, sono solo moduli per inserire i parametri delle ricerche da effettuare, e quindi cambiano di rado. L'accesso molto frequente a pagine di questo tipo, quasi sicuramente registrate nella *cache* del sistema locale dell'utente, in questo caso non verrebbe registrato, anche se meriterebbe di esserlo perché testimonia l'interesse dell'utenza per una specifica funzione.

URL parametrizzate

In un sito Web che serve ad accedere a una base dati, un caso estremamente realistico, ovviamente i risultati di una ricerca non possono costituire un file da scaricare: il sistema non può ospitare una pagina per ogni possibile ricerca effettuata! Già dai primi anni '90 fu affiancata al protocollo HTTP la specifica CGI (Common Gateway Interface, RFC 3875), allo scopo di facilitare la creazione dinamica di pagine Web in base all'input fornito dall'utente, attraverso l'invocazione di programmi residenti sul server. Più tardi sono state introdotte alternative alla CGI, sostanzialmente allo stesso scopo, come le tecnologie Java Servlet, JSP (Java Server Pages), ASP (Active Server Pages), e molte altre.
Il problema con simili tecnologie, pur fondamentali, è che il passaggio dei parametri immessi dall'utente e che servono a produrre, ad esempio, pagine dinamiche contenenti i risultati di una ricerca, può avvenire, fra gli altri, secondo due metodi: GET e POST. Restando nel contesto delle ricerche bibliografiche, vediamo la differenza fra i due metodi.
Nel metodo GET, il programma che effettua la ricerca e crea la pagina con i risultati corrisponde a una certa URL, e i parametri che gli servono sono semplicemente *accodati* alla URL come in questo esempio

http://un.qualche.opac/ricerca.pl?autore=platone&titolo=repubblica

Il programma, in questo caso, si chiama *ricerca.pl*, e gli sono stati trasmessi i parametri *autore* e *titolo*, i cui rispettivi valori sono *platone* e *repubblica*. In modo analogo potrebbero essere trasmessi altri parametri.

Questa URL sarà registrata dal server Web del sito http://un.qualche.opac esattamente come sopra, e quindi, sapendo come possono presentarsi i parametri in queste URL, è possibile ricavare dai *log* informazioni circa le ricerche effettuate. Va detto subito che nessun programma standard di analisi dei *log* fa qualcosa del genere, ma almeno i *log* contengono i dati interessanti, ed è quindi possibile scrivere programmi di analisi per estrarre questi dati.
Questo non è possibile col metodo POST: in questo caso, infatti, le informazioni prima accodate alla URL del programma di ricerca (autore=platone e titolo=repubblica) sono trasmesse al programma stesso come *entità subordinata* alla richiesta, non registrata nei *log* del server Web. Tutto quello che si vedrà nei *log* saranno righe del tipo

... 08:57:53 +0200] "GET **/ricerca.pl** HTTP/1.1" 200 44...

e quindi praticamente tutte uguali. Si potrà solo sapere quante ricerche sono state effettuate, ma nulla si potrà sapere dei parametri utilizzati.
Una situazione molto peggiore, dal punto di vista dell'analisi dei log, è quella in cui il programma è uno solo e l'azione che deve eseguire è essa stessa un parametro. Se il programma è invocato col metodo POST, si avranno righe di *log* da cui non si potrà sapere se l'utente ha fatto una ricerca, ha sfogliato pagine di risultati, ha raffinato una ricerca precedente ecc...
In questi casi, i *log* sono di assai scarsa utilità. Poiché, una volta realizzato un applicativo Web, non è sempre facile modificarlo per evitare una situazione di questo tipo, è opportuno considerare questi aspetti durante l'analisi e la progettazione.

Visitatori e visite

Si è visto che una importante informazione registrata normalmente da un server Web è l'indirizzo IP del *client*, cioè del sistema che, attraverso il protocollo HTTP, ha richiesto al server stesso un qualche file.
Verrebbe spontaneo pensare a questo indirizzo come a un visitatore: in fondo, gli indirizzi IP sono notoriamente univoci, per cui due persone che accedono al sito non possono avere lo stesso indirizzo IP.
Ma anche in questo caso sorgono complicazioni.

Visitatori, prima dei *firewall*

In effetti, quasi tutti gli analizzatori di *log* Web definiscono *visitatore* un indirizzo IP univoco, e riportano tabelle in cui sono conteggiati, ad esempio, gli indirizzi IP distinti registrati nei *log* in un certo periodo di tempo. Da queste tabelle si può pensare di ricavare un'idea approssimativa di quante persone accedono al sistema nell'unità di tempo.
Sarebbe molto bello se fosse così, ma da quando esistono i *firewall* questo quadro idilliaco è del tutto irrealistico.

Firewall

Termine ormai entrato perfino nell'ambito casalingo, sta a indicare una famiglia di sistemi hardware e software usati soprattutto per impedire l'accesso indesiderato a una rete privata. Tipicamente, un *firewall* è configurato in modo che, dall'esterno della rete, siano accessibili soltanto determinati servizi, mentre quelli tipicamente usati all'interno di una LAN (condivisione cartelle e stampanti, ad esempio) restano inaccessibili.

Ma i *firewall* svolgono spesso un'altra importante funzione: la Network Address Translation (NAT). Quando una postazione della LAN accede, ad esempio, a un sito esterno alla LAN, il suo indirizzo IP viene *mascherato*, e la connessione viene invece effettuata dal *firewall* stesso, col proprio indirizzo. Quando il sito risponde, risponde in effetti al *firewall*, che però ha memoria dell'IP della postazione che ha realmente fatto la richiesta, e quindi gira la risposta a quest'ultima.

Il principale vantaggio offerto dalla NAT consiste nel poter oscurare completamente l'esistenza della LAN: nessuno degli indirizzi IP interni appare realmente su Internet. Per questo motivo molte LAN usano tutte gli stessi indirizzi interni (tipicamente, quelli che iniziano con 192.168, oppure con 10.10). Questo non viola l'univocità degli indirizzi IP, perché questi semplicemente non si *vedono* in Internet. Ciò che si vede comunemente al giorno d'oggi sono solo gli indirizzi dei *firewall*.

Visitatori, dopo i *firewall*

Quali problemi pongono i *firewall*, e soprattutto la funzione di NAT che essi normalmente svolgono, nel contesto dell'accesso a un sito Web? Il fatto è che, a causa proprio della NAT, gli indirizzi IP, che gli analizzatori *log* identificano con i visitatori, non corrispondono a delle persone. Vediamo un esempio concreto per fissare le idee.

Consideriamo un ente E, nella cui LAN operano ogni giorno un centinaio di persone, tutte impegnate in attività per le quali consultare il catalogo SBN è un fatto del tutto normale. La LAN è protetta da un *firewall* che effettua la NAT, per cui quando l'utente X effettua una ricerca sull'OPAC SBN, il relativo server Web registra l'indirizzo IP del *firewall* dell'ente E, non quello della postazione dell'utente X. Quando, magari pochi istanti dopo, il collega Y effettua una propria ricerca sull'OPAC SBN, il relativo server Web non può che registrare un'altra volta l'indirizzo IP del *firewall* dell'ente, non quello della postazione dell'utente Y. In conclusione, dal punto di vista dell'OPAC SBN, tutti gli utenti che lavorano nella LAN dell'ente E sono indistinguibili.

La situazione delineata è semplicemente la norma per tutti gli uffici, spesso anche di piccole dimensioni. Stando così le cose, come dare un senso al concetto di visitatore basato sugli indirizzi IP?

Già dai primi anni '90 del XX secolo, alcuni sviluppatori di sistemi di analisi

dei *log* Web hanno rinunciato a cercare di ricavare informazioni di questo genere. Altri sistemi, invece, continuano anche oggi a presentare questi discutibili numeri.

Visite

Concetto strettamente collegato a quello di visitatore, la *visita* a un sito Web è definita in modo ancora più arbitrario e inaffidabile. I prodotti di analisi dei *log* definiscono *visita* un insieme di accessi da uno stesso indirizzo IP localizzati nel tempo, cioè preceduti e seguiti da un certo periodo di inattività (mancanza di accessi) da quell'indirizzo IP. Poiché, come già ricordato sopra, il protocollo HTTP non registra sessioni, ma solo richieste di file, indipendenti le une dalle altre, non sembra esserci altro modo per definire una visita. E d'altra parte la definizione sembra plausibile, ma esaminandola con più attenzione emerge subito la sua intrinseca debolezza.

La durata del periodo di inattività che interviene nella definizione è tipicamente un'ora, in alcuni casi 30 minuti, e comunque è quasi sempre configurabile. Dovrebbe sorgere spontanea la domanda: perché proprio un'ora? Perché non dieci o cinque minuti?

Chiunque abbia lavorato a progetti di catalogazione ha sicuramente effettuato accessi all'OPAC SBN con pause frequentemente inferiori all'ora, almeno in certi periodi. Ma, ancora una volta, sono i *firewall* a demolire questa già debole definizione di visita.

Se gli utenti dell'ente E visto sopra sono un centinaio, e tutti accedono varie volte al giorno all'OPAC SBN, dal punto di vista di certi analizzatori di *log* Web è come se una singola persona (ricordiamo che viene registrato l'indirizzo IP del *firewall*) accedesse quasi di continuo al sistema. A seconda della mole di ricerche effettuate dagli utenti dell'ente E, è possibile che fra un accesso e l'altro non trascorra mai un'ora, o comunque che simili pause, utili a separare una visita dall'altra, siano molto poche. L'analisi dei *log* basata sulla definizione usuale di visita, in queste condizioni, non potrà che conteggiare molte meno visite di quelle effettuate effettivamente dalle singole persone. In altre parole, stante la grande diffusione dei *firewall*, il concetto stesso di visita è da rigettare completamente.

Tempi di risposta

Un'informazione che normalmente non viene registrata da un server Web è il tempo che il server stesso ha impiegato a soddisfare una richiesta, a prescindere dal suo esito. Questa è una sorprendente mancanza, perché una simile informazione sarebbe preziosissima per valutare le prestazioni di un sito Web. In caso, è possibile comunque configurare il server Web in modo che per ogni richiesta sia registrato il tempo che il server ha impiegato a rispondere. Ad esempio, questo è possibile con Apache, il più diffuso server Web al mondo.

Tuttavia, non tutti gli analizzatori di *log* Web prendono in considerazione tale informazione, se presente, e quindi in certi casi questa informazione va elaborata separatamente.

Conclusioni

Gli accessi a un sito Web corrispondono a richieste di trasferimento di file. Quello che, intuitivamente, si definisce come *accesso a una pagina Web* si traduce in realtà in una quantità di singoli accessi, la maggior parte dei quali non contribuiscono alla valutazione del gradimento e dell'uso del sito stesso, per cui bisogna innanzitutto distinguere fra pagine significative, quelle che gli analizzatori di *log* chiamano semplicemente *pagine*, e altri file meno importanti.
A causa dell'onnipresenza dei *firewall*, il concetto di visitatore perde molto del suo valore, dal momento che tutti gli utenti che accedono a un sito Web da una LAN protetta da *firewall* corrispondono a un unico indirizzo IP, per cui il server Web li considera come un unico utente.
Per motivi analoghi, anche il concetto di visita, introdotto surrettiziamente per ovviare alla mancanza di sessioni in HTTP, e già molto debole di per sé stesso, perde quasi ogni valore.
La configurazione del server Web dal punto di vista dei *log* è quindi critica, e non andrebbe data per scontata, perché le impostazioni predefinite di molti server Web sono spesso generiche, in modo da adattarsi a molte situazioni, ma così vengono spesso ignorate informazioni importanti.
Restano comunque informazioni preziose nei *log* Web e nella loro analisi. Ad esempio, il numero di accessi a un modulo per la ricerca bibliografica è sicuramente proporzionato al numero di ricerche effettuate e, se si accetta il fatto che non possiamo sapere da chi, si ha comunque un indice dell'interesse del pubblico verso un determinato strumento.

Common expressions such as "I am connected to such-and-such Website" can be absolutely acceptable and clear in our every day lives, but they can be the source of many a misunderstanding when it comes to Web statistics. Specifically, when I am reading a page visualised by my browser, am I not "connected to the Website" at all. When I access through a browser a Webpage contained in a given site, the browser submits the appropriate request to the Web server, and the server answers by sending over the page in question, or by notifying possible errors. The HTTP protocol does not establish sessions at all, it only opens a connection for each file that has to be transferred and ends it once the transfer has been completed (and is thus different from the FTP protocol, which allows for any number of files to be transferred and for other operations to be carried out in the course of a single session).
For each request that it receives, the Web server records a great deal of information into the appropriate log file. There are many programs, either commercially

or freely available, designed to analyse these log files and extract a wide range of statistical data from them. Web administrators often expect to be able to derive from this data some precious information about how their Websites are used or how popular they are. But the truth is that the interpretation of such data can be considered reliable only under some rather restrictive conditions.
For example, an important information which a Web server would normally record is the IP address of the client, that is to say the address of the system that has requested a file to the server via the HTTP protocol. It would be natural to think of this address as a visitor: after all, IP addresses are notoriously unique, meaning that if two different people access the Website, they cannot have the same IP address. As a matter of fact, most Web log analysers would define a unique IP address as a visitor, and produce tables which, amongst other things, calculate how many different IP addresses were recorded in a log over a given time interval. Hence, one would expect that these tables make it possible to roughly estimate how many people have accessed the system in the course of time. It would be fantastic if it were so, but since the arrival of firewalls this expectation is totally unrealistic. In fact, if a firewall is active, the server can only record in the log file the firewall address, but it can in no way see through the firewall and detect the other IP addresses which are active behind it.
And this is just one of the many difficulties making Web statistics a rather complex business.

Des expressions habituelles telles que "je suis connecté au site x" qui sont tout à fait acceptables et claires dans un contexte quotidien peuvent cependant créer des malentendus dans le contexte des statistiques Web. En particulier, lorsque l'on est en train de lire une page affichée par le browser on n'est pas du tout "connecté au site". Lorsque l'on accède avec le browser à une page Web d'un site, le browser transfère au serveur une demande spécifique à laquelle ce dernier répond en envoyant la page en question ou en signalant des erreurs éventuelles. Le protocole HTTP n'établit pas des sessions mais ouvre une connexion pour chaque fichier à transférer et la ferme dès que le transfert a terminé (différemment du protocole FTP qui, au long d'une session, permet de transférer un nombre quelconque de fichiers et d'effectuer aussi d'autres opérations).
Pour chaque demande reçue, un serveur Web enregistre de nombreuses informations dans un fichier de log spécifique. Il existe de nombreux programmes, aussi bien commerciaux que gratuits, qui analysent les fichiers de log en y puisant une vaste série de données statistiques. Souvent les responsables administratifs pensent pouvoir extraire de ces données des informations précieuses sur l'utilisation et appréciation de leurs sites Web. Mais en réalité, l'interprétation de ces données n'est valable qu'à la lueur de considérations très restrictives.
Un exemple: une information importante enregistrée normalement par un serveur Web est l'adresse IP du client, soit du système qui a demandé au serveur un fichier à travers le protocole HTTP. Il semblerait évident de penser à cette adresse comme à un visiteur car il est notoire que les adresse IP sont univoques

et donc deux personnes accédant au site ne peuvent pas avoir la même adresse IP. En effet, presque tous les analyseurs de log Web définissent visiteur une adresse IP univoque et font des tableaux où, entre autre, sont comptées les adresses IP enregistrées dans les log à une période donnée. On pense que l'on peut extraire de ces tableaux une idée approximative du nombre de personnes qui ont accédé au système pendant cette période. Cela serait très beau s'il en était puisque depuis l'existence du firewall ce cadre idyllique est tout à fait irréel. Dans ce cas, en effet, le serveur ne peut enregistrer dans le fichier de log que l'adresse du firewall sans pouvoir voir toutes les autres adresses IP actives derrière le firewall lui-même.
Cela constitue l'un des nombreux problèmes qui font du thème des statistiques Web un sujet plutôt complexe.

RIFERIMENTI BIBLIOGRAFICI

Analog 6.0. *How the Web works*, http://www.analog.cx/docs/Webworks.html.

Apache HTTP Server Documentation, http://httpd.apache.org/docs/.

AWStats, http://www.awstats.org.

Doug Linder. *Interpreting WWW Statistics*, http://www.ario.ch/etc/Webstats.html.

The Internet Engineering Task Force. *Hypertext Transfer Protocol – HTTP/1.1*, http://www.ietf.org/rfc/rfc2616.txt.

The Internet Engineering Task Force. *The Common Gateway Interface (CGI) Version 1.1*, http://www.ietf.org/rfc/rfc3875.

Siti Web

Statistica dell'OPAC dell'Indice SBN, http://opac.stats.sbn.it/awstats/awstats.pl?config=opac.sbn.it&lang=it.

Statistica di Internet Culturale, http://www.internetculturale.it/aws/awstats.pl.