

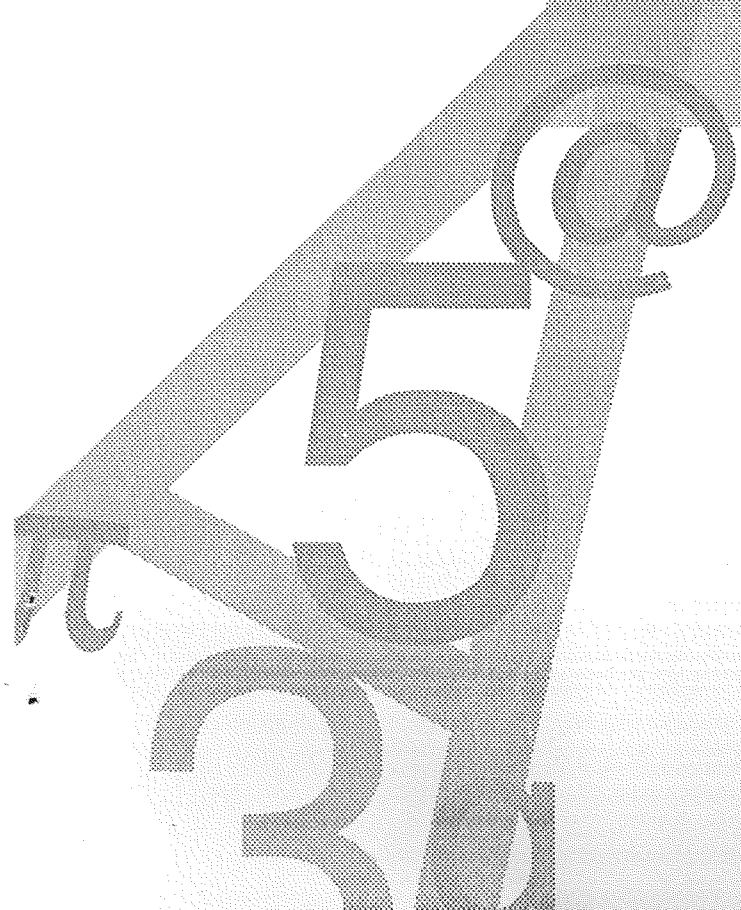
Consiglio Nazionale delle Ricerche

**Rilevazione ed analisi dei dati
di accesso ad un sistema WWW**

A. Ceccarelli e M. Di Giacomo

CNUCE: C96 - 13

CNUCE



Rilevazione ed analisi dei dati di accesso ad un sistema WWW

Alfredo Ceccarelli (*CNUCE - Istituto del CNR*)

Mariella Di Giacomo (*Consorzio Pisa Ricerche*)

Rapporto Interno C96-13

Pisa, Aprile 1996

Indice

1. Introduzione	1
2. World Wide Web	2
2.1 Configurazione di un server WWW	4
2.2 I file di configurazione del server	8
2.3 I file di log del server	9
2.3.1 Descrizione dei dati di log	11
3. Descrizione del programma	13
3.1 Il Cliente	14
3.2 Il Server	15
4. Rilevamento di un campione	17
5. Conclusioni	19
Appendice A	20
Appendice B	21
Bibliografia	31

1. Introduzione

Il rapporto descrive lo strumento creato per rilevare le informazioni relative agli accessi degli utenti verso il server *World Wide Web* (WWW). Sulla macchina funzionante da server è definito anche un motore di ricerca per localizzare i documenti presenti nella Banca Dati che costituisce il sistema informativo degli utenti.

Il sistema distribuito utilizza, per l'accesso alla Banca Dati il modello cliente server (modello asimmetrico di comunicazione in cui esiste un processo server in grado di fornire i servizi, ed altri processi ai clienti che li richiedono), per accedere alle informazioni. Esso può essere quindi pensato come un sistema integrante un processo cliente, un processo server ed il meccanismo che permette loro di comunicare.

In questo rapporto viene fatto riferimento al server WWW fornito dal consorzio W3C e, come piattaforma di riferimento viene utilizzata un *Personal Computer Pentium* con il sistema operativo Linux.

Il Capitolo 2 contiene le specifiche per la configurazione di un server WWW.

Nel Capitolo 3 viene descritto il formato dei dati, registrati dal server WWW (riferenti gli accessi degli utenti) utilizzati per effettuare le statistiche di accesso al medesimo.

Il Capitolo 4 fornisce una descrizione formale dei programmi cliente e server realizzati.

Nel Capitolo 5 viene mostrata una rappresentazione grafica relativa al campione rilevato.

2. World Wide Web

Il sistema *World Wide Web* (WWW) è un servizio di *Internet* che consente di accedere alle informazioni contenute nella rete telematica che collega milioni di computer (stazioni di lavoro, *personal computer*, *Apple Macintosh*) sparsi per il mondo.

Le informazioni contenute in tale rete sono organizzate secondo il modello ipertestuale. Questo tipo di organizzazione consente di muoversi nella rete semplicemente passando da un documento all'altro senza specificare nomi o indirizzi dei nodi della rete.

Un ipertesto è un metodo per presentare le informazioni in cui le parole selezionate nel testo possono essere espansive ottenendo per esse ulteriori informazioni. Cioè tali parole sono legami ad altri documenti, che possono essere testi, file, figure o altre forme di informazioni elettroniche.

In un ipertesto è possibile scegliere un proprio percorso di lettura. Questo significa passare da un punto ad un altro, ottenere informazioni, tornare indietro, saltare ad un altro argomento e magari fare annotazioni e aggiungere nuove parti al documento. La tecnologia WWW è uno degli strumenti più flessibili per navigare nella rete *Internet*.

Il sistema WWW è stato sviluppato al CERN di Ginevra per permettere ai fisici in tutto il mondo di comunicare in modo più efficiente usando l'informazione ipertestuale. Il primo *software* è stato rilasciato pubblicamente nell'agosto 1991. Nel febbraio 1993 il centro di ricerca della NCSA (*National Center for Supercomputing Applications at the University of Illinois*) ha rilasciato la prima interfaccia grafica (*browser*) per il sistema.

Successivamente, dalla NCSA sono stati realizzati i serveri WWW per le diverse piattaforme.

Il servizio di informazioni WWW è basato sul paradigma cliente-server ed essendo un servizio di Internet utilizza la famiglia dei protocolli TCP/IP per la comunicazione fra i clienti ed i serveri. La fig.1 fornisce lo schema di interazione specificato.

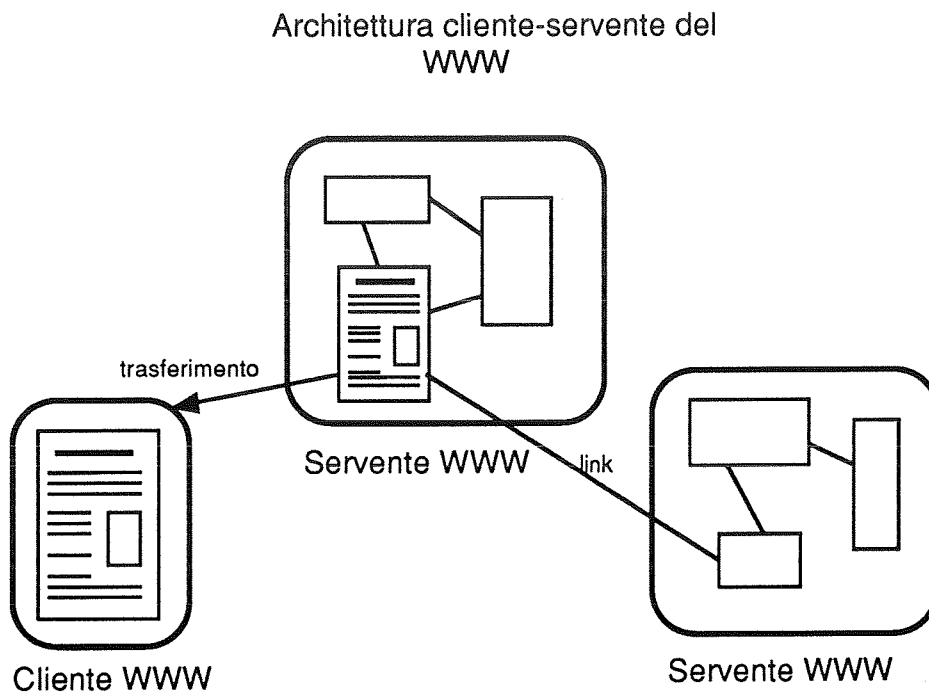


Fig. 1 - Architettura cliente-servere del www

Le informazioni risiedono su sistemi detti serveri WWW e possono consistere di una vasta gamma di dati multimediali, ovvero testo, suoni, immagini e persino spezzoni video. Il server è il *software* che gestisce e rende disponibili tali informazioni nella rete. Esistono molti serveri WWW, per i vari sistemi operativi: Unix, Linux, Windows, OS/2 e MacOS.

Il cliente, *software* usato da un utente per accedere alle informazioni gestite dal server WWW, nella terminologia viene comunemente detto *browser*. I clienti più diffusi sono: *Netscape* e *Mosaic*. Entrambi sono supportati dal sistema operativo UNIX con il sistema a finestre *X-WINDOWS*, dal *Microsoft Windows* e dal sistema operativo residente su piattaforma *Macintosh*, ma si consiglia l'uso di *Netscape* per una migliore visualizzazione grafica.

In questo rapporto viene fatto riferimento al server WWW sviluppato al CERN e, come piattaforma di riferimento viene utilizzato un *Personal Computer Pentium* con il sistema operativo Linux.

Viene enfatizzata la parte relativa al server in quanto è la componente che provvede direttamente alla registrazione delle informazioni che costituiscono l'oggetto della nostra analisi.

2.1 Configurazione di un server WWW

Ogni server necessita di uno o più file di configurazione, contenenti le informazioni necessarie per rendere lo stesso flessibile alle modifiche di tali informazioni, evitando la ricompilazione dei sorgenti.

Durante il processo di installazione del server, sono creati alcuni file e directory che sono parte integrante del server e costituiscono il supporto per il funzionamento dello stesso.

Le directory di lavoro, così chiamate, sono create nella directory dove è installato il server. La lista che segue, indica

quelle usate dall'amministratore per controllare il funzionamento e valutare le richieste di accesso al server: "icons", "logs", "support", "cgi-bin", "cgi-src", "conf". Si fornisce ora una breve descrizione delle stesse:

- "icons" contiene immagini che possono essere utili da includere in documenti residenti sul server;
- "logs" indica la locazione per la registrazione dei messaggi di errore e gli accessi dei clienti verso il server;
- "support" in questa directory si trovano alcuni programmi di utilità;
- "cgi-bin" contiene i programmi, chiamati anche *script Common Gateway Interface (CGI)*, che vengono eseguiti sul server.

Il testo che viene interpretato da ogni cliente WWW può essere contenuto in un file oppure essere dinamicamente generato da un programma che viene eseguito sulla macchina che ospita il server WWW, eventualmente in funzione dell'input proveniente dalla compilazione di una *form* (scheda o modulo) da parte dell'utente.

Il meccanismo che consente al server di ricevere i dati dal cliente e di eseguire i programmi necessari ad elaborarli si chiama

Common Gateway Interface, da cui il termine *script* CGI. L'interfaccia CGI si compone di due parti: ricezione dei dati ed esecuzione dei programmi. La ricezione dei dati è standardizzata, cioè avviene in modo indipendente dal tipo di macchina e di sistema operativo su cui sono eseguiti i clienti e i server, mentre l'esecuzione dei programmi è dipendente dal sistema operativo in quanto ogni sistema gestisce questa fase in modo diverso.

Inoltre, il server deve comunicare con i programmi di tipo CGI per inviar loro i dati ricevuti dal cliente, e anche questa fase è dipendente dal sistema operativo. Di conseguenza i meccanismi di ricezione dati sono diversi per sistemi operativi differenti.

Il termine *script* fa riferimento alla possibilità di scrivere i programmi nel linguaggio di programmazione delle *shell* disponibili sotto Unix ma in pratica essi possono essere realizzati utilizzando un qualsiasi linguaggio di programmazione. Nel sistema operativo Unix i linguaggi di programmazione più utilizzati sono: Perl, csh, sh, C; in Windows 3.11 si può utilizzare *Visual Basic*, e su Macintosh si può programmare in *Applescript*;

- “cgi-src” contiene i sorgenti dei programmi *script* CGI che sono nella directory “cgi-bin.”.

Il contenuto della directory può essere usato come utile riferimento per scrivere i propri file di script. Alcuni campi di applicazione di questo tipo di file sono: inviare un messaggio di posta elettronica (*e-mail*), processare dati di input (interrogare basi di dati, rispondere a questionari, scegliere prodotti da un catalogo) o chiamare un altro programma;

- “conf” la directory contiene i file di configurazione del server descritti nella sezione 2.2.

Tra le varie opportunità che il server offre c'è quella di fornire il servizio *proxy*, cioè filtrare tutte le richieste effettuate dal cliente, con la capacità ulteriore di effettuare una copia locale dei documenti recuperati (*caching*). Se il server è abilitato al *caching* ogni volta che gli arriva una richiesta viene controllato se il documento oggetto di tale richiesta è già a disposizione del server. In tal caso il documento viene restituito al cliente, evitando il collegamento con la macchina remota indicata nell'URL (*Universal Resource Locator*), altrimenti lo recupera presso il server remoto e ne fa una copia da utilizzare per eventuali richieste successive.

2.2 I file di configurazione del server

Quando viene attivato il server, esso legge i file di configurazione e li carica in memoria, ove rimangono finché il server è attivo. Il principale file di configurazione è "httpd.conf".

Il server W3C può richiedere solo questo file di configurazione, mentre nel server NCSA sono presenti anche altri file come: "srm.conf" (fornisce informazioni sulla locazione ed il tipo di documenti sul server), "access.conf" (definisce quali utenti possono accedere al server e a quali directory), "mime.types" (associa i nomi di file ai tipi di documento).

"Httpd.conf" è il file di configurazione dove sono specificate, tramite parole chiave chiamate "direttive", le informazioni di controllo del server. Tra le informazioni in esso contenute, vi sono: la directory radice per il server, a partire dalla quale si risolvono tutti i riferimenti relativi (*ServerRoot*), ed il nome completo della macchina sulla quale il processo server è in esecuzione (*HostName*), comprensivo del dominio. Altra informazione importante è la porta sulla quale il server si pone in attesa di ricevere le richieste di connessione (*Port*).

All'interno del file si trovano anche le direttive per controllare l'accesso ai documenti privati e quelle per i file dove il sistema memorizza le informazioni relative agli errori e agli accessi dei clienti verso il server.

Nell'appendice A è riportato un esempio del file di configurazione per il server W3C che lavora come un server standard.

2.3 I file di log del server

Il server WWW può registrare le richieste di accesso, inviate dai vari clienti, su appositi file chiamati file di registrazione o di *log*. Tale registrazione viene abilitata inserendo nel file di configurazione una linea di controllo del seguente tipo:

```
AccessLog percorso_file
```

dove il termine "percorso_file" rappresenta il percorso assoluto o relativo (a partire dalla directory radice del server) del nome del file in cui si desidera memorizzare le informazioni e terminante con il prefisso che il nome del file di registrazione deve avere. Tale prefisso verrà concatenato al nome rappresentante la data di creazione del file stesso.

Le informazioni memorizzate in questo file di log, sono relative ai documenti trasferiti dal server. Per ogni trasferimento vengono registrati la data, l'ora, il nome dell'host remoto o l'indirizzo IP e il nome del documento richiesto.

Le informazioni associate al file costituiscono la base primaria per l'analisi sull'utilizzo del server.

E' possibile disabilitare la registrazione delle richieste provenienti da determinate macchine utilizzando la linea di controllo:

```
NoLog stringa
```

dove il termine "stringa" indica l'indirizzo IP di un sistema o di un insieme di sistemi.

Nel caso in cui il server WWW espleti la funzionalità di *proxy* è possibile registrare le richieste pervenute per tale servizio in un file distinto tramite la direttiva:

ProxyAccessLog percorso_file

Il termine "percorso_file" specifica il percorso ed il nome del file di memorizzazione e può essere: assoluto o relativo alla directory radice del server, come specificato con la direttiva *ServerRoot*. Nel caso in cui il server funzioni sempre in qualità di *proxy* è possibile registrare gli accessi alle informazioni contenute nella propria *cache* tramite la linea di controllo:

CacheAccessLog percorso_file

Il termine "percorso_file" ha lo stesso significato indicato nella direttiva precedente.

Oltre alla possibilità di effettuare la registrazione delle informazioni relative alle richieste degli utenti, è possibile registrare su di un file le informazioni relative alle richieste errate tramite la linea di controllo:

ErrorLog percorso_file

Il termine "percorso_file" specifica, ancora una volta, il percorso assoluto o relativo ed il nome del file di memorizzazione.

Nel file specificato dalla direttiva sono memorizzate le informazioni sugli errori, come: un documento che non è stato

trovato, timeout dovuto a bassa comunicazione, connessione che è stata interrotta, errore di script, invalido file di configurazione.

Le informazioni riportate nel file relativo alla registrazione degli errori costituiscono, perciò, la fonte primaria per capire se qualcosa nel sistema non funziona.

2.3.1. Descrizione dei dati di Log

Le informazioni contenute nei file di registrazione sono generate utilizzando un formato di memorizzazione standard per tutti i serveri WWW, allo scopo di rendere possibile la realizzazione di programmi di statistiche indipendentemente dal server che ha effettuato le registrazioni.

Le informazioni relative agli accessi dei clienti verso un server WWW sono memorizzate in un unico tipo di record. Il formato del record, indicato nel file di configurazione dalla linea di comando "*LogFormat Common*" consiste di una serie di attributi, elencati di seguito, ognuno separati dal carattere "spazio":

nome del sistema remoto

Tale attributo indica il nome del sistema remoto (o indirizzo IP se non è possibile eseguire la risoluzione del nome da parte del DNS o se nel file di configurazione la linea di controllo *DNSLookup* non è attivata) dal quale è pervenuta la richiesta di accesso;

rfc931	Indica il nome dell'utente remoto che ha effettuato la richiesta;
autenticazione	L'attributo indica il nome che l'utente ha utilizzato per autenticarsi durante la richiesta;
[data]	Tale attributo indica la data e l'ora in cui è pervenuta la richiesta, il tutto racchiuso fra parentesi quadre e nel formato standard con cui viene rappresentata l'ora locale oppure specificando la direttiva " <i>LogTimeLocalTime</i> GMT" nel file di configurazione del server. Oltre a tale possibilità si può fornire al server una direttiva direttiva in cui si specifica un diverso formato di rappresentazione della data e dell'ora. La linea di controllo da utilizzare è " <i>LogFileDateExt</i> modello", dove il termine "modello" rappresenta il formato che si desidera utilizzare;
richiesta	L'attributo indica la richiesta, espressa tramite l'URL, che il cliente ha inviato. Tale informazione è racchiusa tra doppi apici;
stato	L'attributo indica l'esito della richiesta (esso è un valore appartenente all'insieme dei valori restituiti dalle funzioni appartenenti al protocollo HTTP);

dimensione del documento trasferito

L'attributo indica la lunghezza in byte del documento che viene visualizzato al cliente, relativamente alla richiesta inviata.

3. Descrizione del programma

Il programma si compone di una parte cliente che viene eseguita sulla macchina locale e di un servente (*script* CGI) eseguito sulla macchina remota. In Appendice B viene riportato il codice relativo.

Il testo interpretato dal cliente per l'analisi dei dati di log, si avvale del meccanismo CGI tramite il quale è possibile fornire i dati al servente ed eseguire i programmi necessari ad elaborarli.

Gli *script* acquisiscono il loro input prevalentemente dalle variabili di ambiente o dallo standard input (gli *script* per le interrogazioni ricevono le chiavi dagli argomenti forniti da linea di comando). La variabile di ambiente più importante è "QUERY_STRING", che contiene la stringa dell'interrogazione codificata, cioè ciò che segue il carattere "?". Tale stringa può contenere caratteri speciali come gli spazi, le linee nuove (codificati nella rispettiva notazione esadecimale) ed i caratteri come il "+", "=", "&". Il contenuto di tale variabile può essere facilmente controllato utilizzando il programma "cgiparse".

Gli *script* restituiscono il loro risultato allo standard output e quindi direttamente al cliente o la locazione del documento risultante. Uno *script* deve iniziare con una linea contenente il

tipo di documento che quindi crea una intestazione MIME che specifica il formato dei dati che seguiranno ed una linea vuota essenziale che indica la fine dell'intestazione e l'inizio dei dati.

I dati possono essere passati al server in due diverse modalità. Nel primo metodo i dati vengono concatenati all'URL dello script mentre con il secondo metodo vengono mandati direttamente al server. In genere conviene utilizzare il secondo metodo in quanto meno soggetto a limitazioni sul numero di dati che si possono passare allo script.

La richiesta di input e la successiva generazione di pagine il cui contenuto dipende dai dati precedentemente forniti è consentita tramite la specifica scheda o *form*. All'interno di un documento HTML le *form* sono marcate dal tag `<form> </form>`.

3.1 Il Cliente

< la direttiva FORM indica l'URL dello *script* da eseguire e la modalità di ricezione dei dati. Con la modalità di ricezione specificata i dati vengono concatenati all'URL in modo tale che il server li estrae da esso e li passa allo *script* >

< viene scelto il nome di un mese, selezionato dalla lista fornita, del quale si desidera leggere i dati delle statistiche >

< con la successiva lista viene scelto l'anno di interesse >

< i dati della *form* vengono inviati al server >

3.2 Il Servente <statistics_script>

```

if <(anno corrente è uguale all'anno richiesto)> then
  if <(mese corrente è uguale al mese richiesto) > then
    {<se esiste viene rimosso il file temporaneo delle
      statistiche per il mese corrente>
      <log_script> <creazione del file temporaneo
        delle statistiche per il mese corrente>
      <visualizzazione dei risultati al cliente>
    }
  if < (mese corrente<mese richiesto) > then
    {<si controlla se esiste il file definitivo delle
      statistiche per quel mese>
      if < file definitivo esiste > then
        {<visualizza i risultati definitivi per quel
          mese al cliente>
        }
      else
        {< crea il file definitivo delle statistiche
          per il mese precedente>
          < visualizza al cliente i risultati
            definitivi per quel mese>}
    }
  if <(mese richiesto dello stesso anno > del corrente)>
  then
    { < visualizza un messaggio con cui si informa il
      cliente che non è possibile effettuare le statistiche
      per tale mese>}
  if < (anno richiesto > anno corrente) > then

```

```

    { < visualizza un messaggio con cui si informa il cliente che
      non è possibile effettuare le statistiche per tale anno > }
  if < (anno richiesto < anno corrente) > then
    { < si controlla se per tale anno è stato possibile effettuare
      le statistiche >
      if < esiste il file delle statistiche per tale mese > then
        { < visualizza i dati per tale mese > }
      else
        { < se è possibile creare il file definitivo delle
          statistiche lo si crea e si visualizzano i dati al
          cliente > }
    }
}

```

Servernte <log_script>

Questa è una routine interna alla parte servernte.

```

for < ogni file di log appartenente allo stesso mese >
  { < si controlla quante volte un utente ha acceduto al
    servizio WWW, si ricorda la data del primo
    accesso in tale mese e si somma il numero di
    accessi >
    < tale lista viene ordinata rispetto al nome degli
      utenti e costruita in modo da essere interpretata
      dai clienti WWW >
  }

```

4. Rilevamento di un campione

Il rilevamento del campione è stato realizzato per un particolare sistema distribuito. Il criterio utilizzato può comunque essere applicato a qualsiasi sistema distribuito WWW.

Il campione si riferisce alla misurazione effettuata nei mesi di Gennaio, Febbraio e Marzo 1996 (rispettivamente riportato nelle Fig. 2, Fig. 3, Fig. 4). Ogni grafico è stato realizzato riportando in ascissa il nome dell'utente ed in ordinata il numero di accessi effettuati, complessivamente da ciascuno, nel mese relativo.

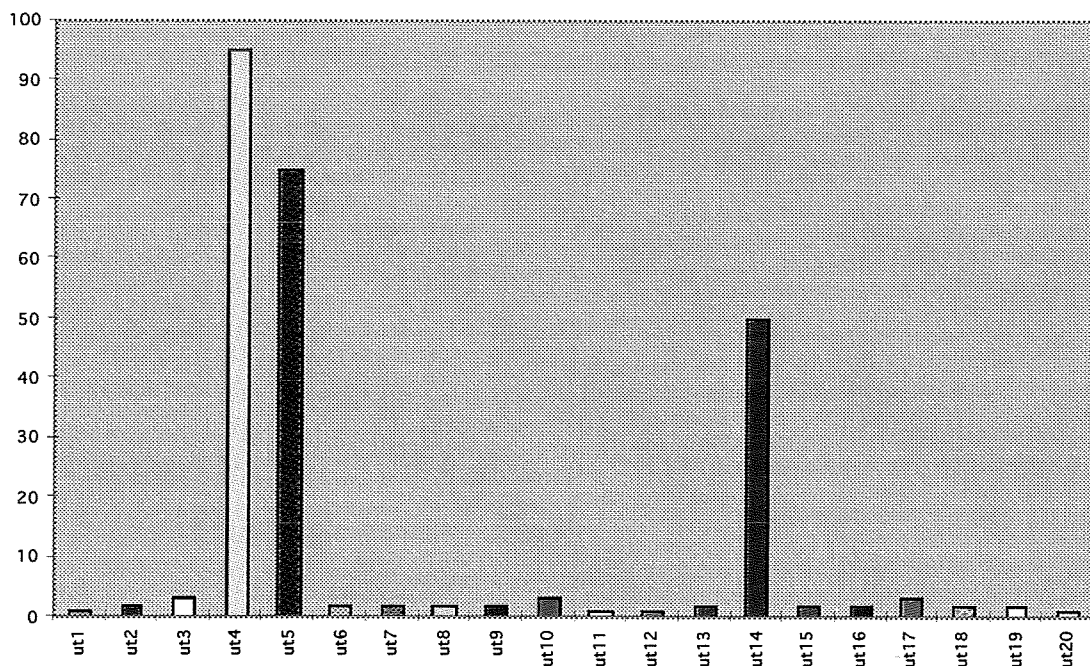


Fig. 2 - Accessi al server nel mese di Gennaio

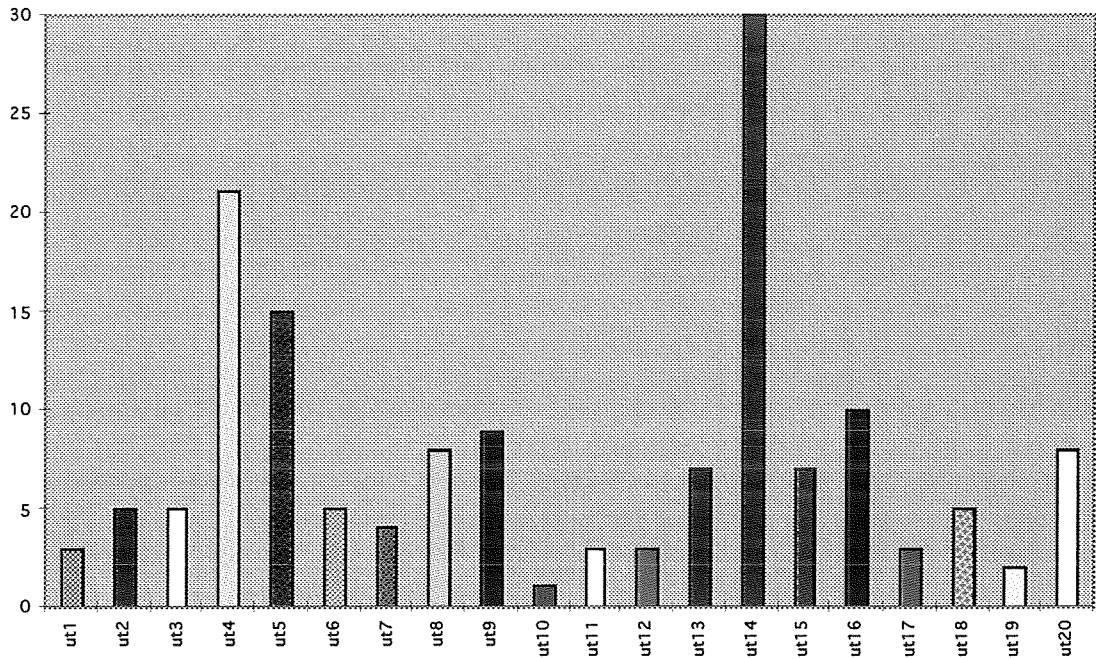


Fig. 3 - Accessi al server nel mese di Febbraio

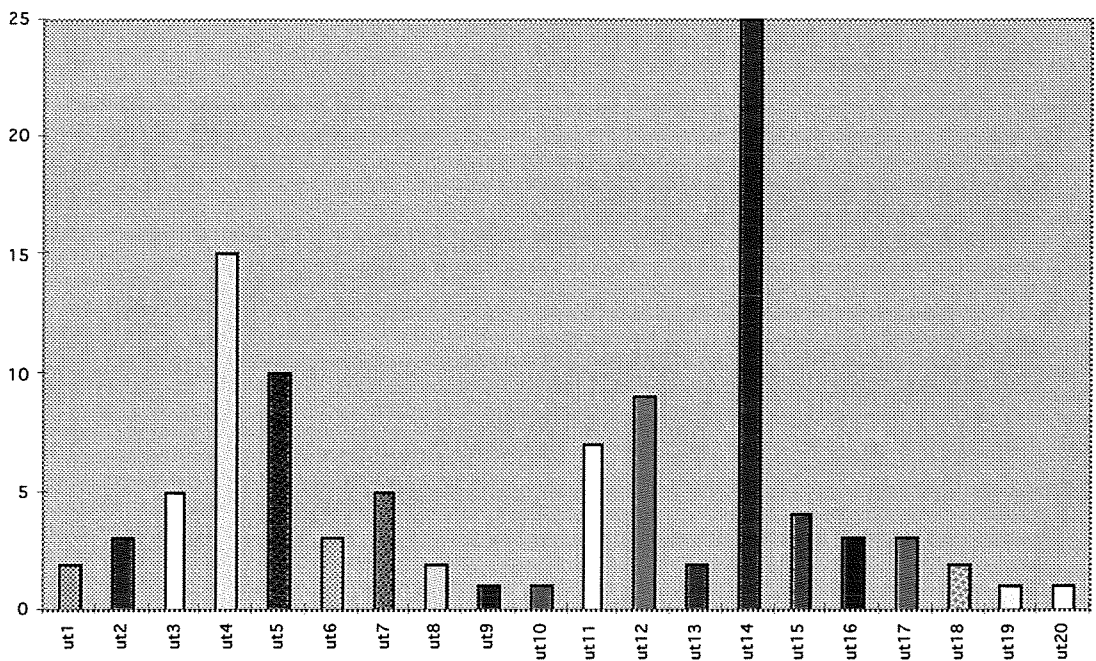


Fig. 4 - Accessi al server nel mese di Marzo

5. Conclusioni

Nel presente rapporto è stato presentato un esempio di *Common Gateway Interface*. L'interfaccia consente al server WWW di ricevere le informazioni, provenienti dai clienti *Netscape*, per eseguire il programma realizzato ed elaborare i file di registrazione contenenti i dati sugli accessi degli utenti verso determinati sottoalberi contenenti le informazioni gestite dal server WWW e WAIS.

```

SED_CHANGE_MONTH="sed_change_month"
FILES=""
TITLE=""
echo Content-type:text/html
echo
eval `$CGIPARSE -form`
REQUESTED_YEAR=$FORM_year_name
REQUESTED_YEAR_DIR="$REQUESTED_YEAR/"

#Viene di seguito valutata la data odierna
#
CURRENT_DATE=`date | awk '{ print $3, $2, $6 }`
CURRENT_YEAR=`date | awk '{ print $6 }`
CURRENT_MONTH=`date | awk '{ print $2 }`
case $FORM_month_name in

"Gennaio") REQUESTED_MONTH="Jan";
            POSTFIX="Jan";
            FILES="$LOGS_DIR/logs.Jan*";
            PROGRAM="log_script";
            TITLE=" Traffico nel mese di Gennaio ";;

"Febbraio")  REQUESTED_MONTH="Feb";
            POSTFIX="Feb";
            PROGRAM="log_script";
            FILES="$LOGS_DIR/logs.Feb*";
            TITLE=" Traffico nel mese di Febbraio ";;

"Marzo")  REQUESTED_MONTH="Mar";
            POSTFIX="Mar";
            PROGRAM="log_script";
            FILES="$LOGS_DIR/logs.Mar*";
            TITLE=" Traffico nel mese di Marzo ";;

```

```
"Aprile") REQUESTED_MONTH="Apr";
          POSTFIX="Apr";
          PROGRAM="log_script";
          FILES="$LOGS_DIR/logs.Apr*";
          TITLE=" Traffico nel mese di Aprile ";;

"Maggio") REQUESTED_MONTH="May";
          POSTFIX="May";
          FILES="$LOGS_DIR/logs.May*";
          PROGRAM="log_script";
          TITLE="Traffico nel mese di Maggio ";;

"Giugno") REQUESTED_MONTH="Jun";
          POSTFIX="Jun";
          FILES="$LOGS_DIR/logs.Jun*";
          PROGRAM="log_script";
          TITLE=" Traffico nel mese di Giugno ";;

"Luglio") REQUESTED_MONTH="Jul";
          POSTFIX="Jul";
          PROGRAM="log_script";
          FILES="$LOGS_DIR/logs.Jul*";
          TITLE=" Traffico nel mese di Luglio ";;

"Agosto") REQUESTED_MONTH="Aug";
          POSTFIX="Aug";
          PROGRAM="log_script";
          FILES="$LOGS_DIR/logs.Aug*";
          TITLE=" Traffico nel mese di Agosto ";;

"Settembre") REQUESTED_MONTH="Sep";
          POSTFIX="Sep";
          PROGRAM="log_script";
          FILES="$LOGS_DIR/logs.Sep*";
          TITLE=" Traffico nel mese di Settembre ";;
```



```

"Ottobre") REQUESTED_MONTH="Oct";
        POSTFIX="Oct";
        PROGRAM="log_script";
        FILES="$LOGS_DIR/logs.Oct*";
        TITLE=" Traffico nel mese di Gennaio ";;

"Novembre") REQUESTED_MONTH="Nov";
        POSTFIX="Nov";
        PROGRAM="log_script";
        FILES="$LOGS_DIR/logs.Nov*";
        TITLE=" Traffico nel mese di Novembre ";;

"Dicembre") REQUESTED_MONTH="Dec";
        POSTFIX="Dec";
        PROGRAM="log_script";
        FILES="$LOGS_DIR/logs.Dec*";
        TITLE=" Traffico nel mese di Dicembre ";;

esac
#
DIGIT_REQUESTED_MONTH=`echo $REQUESTED_MONTH | sed -f
$SCRIPT_DIR/$SED_CHANGE_MONTH`;
DIGIT_CURRENT_MONTH=`echo $CURRENT_MONTH | sed -f
$SCRIPT_DIR/$SED_CHANGE_MONTH`;
#
if test $PROGRAM = "log_script"; then
    # Si controlla se l'anno coincide

    if [ $CURRENT_YEAR -eq $REQUESTED_YEAR ]; then
        # Si controlla se il mese, nello stesso anno, coincide
        if [ $DIGIT_REQUESTED_MONTH -eq $DIGIT_CURRENT_MONTH ]; then

            echo "<title> $FORM_month_name </title>"
            echo "<p><center><h2>"

```

```

echo "$TITLE</h2> <p>"
# Si rimuove il file temporaneo delle statistiche per il mese corrente se
esiste
if [ -s $OUTPUT_DIR$PREFIX$REQUESTED_MONTH ]; then
rm $OUTPUT_DIR$PREFIX$REQUESTED_MONTH
fi

# Si visualizza il file temporaneo delle statistiche creato per il mese
corrente

#cat ` $SCRIPT_DIR/$PROGRAM $FILES`;
$SCRIPT_DIR/$PROGRAM $FILES;
cat $OUTPUT_DIR$PREFIX$POSTFIX;

#terminazione
# if [ $DIGIT_REQUESTED_MONTH -eq $DIGIT_CURRENT_MONTH ]
fi

# Si controlla se il mese pervenuto, nello stesso anno, e' minore del corrente
if [ $DIGIT_REQUESTED_MONTH -lt $DIGIT_CURRENT_MONTH ]; then

# Si controlla se il file definitivo delle statistiche per quel mese esiste
if [ -s
$OUTPUT_DIR$REQUESTED_YEAR_DIR$END_PREFIX$REQUESTED_MON
TH ]; then
echo "<title> $FORM_month_name </title>"
echo "<p><center><h2>"
echo "$TITLE</h2> <p>"
#echo " Il nome del file e' :
$OUTPUT_DIR$REQUESTED_YEAR_DIR$END_PREFIX$REQUESTED_MON
TH "
cat
$OUTPUT_DIR$REQUESTED_YEAR_DIR$END_PREFIX$REQUESTED_MON
TH

```

```

else
    # Si e' nel caso in cui il file di statistiche definitivo di un mese
    # precedente, nello stesso anno, non esiste e va creato

    # Si rimuove il file temporaneo delle statistiche per il mese precedente se
    esiste

    if [ -s $OUTPUT_DIR$PREFIX$REQUESTED_MONTH ]; then
        rm $OUTPUT_DIR$PREFIX$REQUESTED_MONTH
    #terminazione
    # if [ -s $OUTPUT_DIR$PREFIX$REQUESTED_MONTH ]; then
    fi
    # Si crea il file temporaneo delle statistiche creato per il mese precedente
    $SCRIPT_DIR/$PROGRAM $FILES;

    # Si rende il file temporaneo creato definitivo

    mv $OUTPUT_DIR$PREFIX$REQUESTED_MONTH
    $OUTPUT_DIR$REQUESTED_YEAR_DIR$END_PREFIX$REQUESTED_MON
    TH
    echo "<title> $FORM_month_name </title>"
    echo "<p><center><h2>"
    echo "$TITLE</h2> <p>"

    # Si visualizza il file definitivo per il mese richiesto
    cat
    $OUTPUT_DIR$REQUESTED_YEAR_DIR$END_PREFIX$REQUESTED_MON
    TH
    #terminazione if [ -s
    $OUTPUT_DIR$REQUESTED_YEAR_DIR$END_PREFIX$REQUESTED_MON
    TH ]; then
    fi
    #terminazione if [ $DIGIT_REQUESTED_MONTH -lt
    $DIGIT_CURRENT_MONT ]; then

```

```

fi
# Si controlla se il mese pervenuto, nello stesso anno, è maggiore del
corrente
if [ $DIGIT_REQUESTED_MONTH -gt $DIGIT_CURRENT_MONTH ]; then
    echo "<p><center>"
    #echo "<p><center><h2>"
    #echo "$TITLE</h2><p> "
    echo "Non e' possibile effettuare le statistiche"

# terminazione if [ $DIGIT_REQUESTED_MONTH -gt
$DIGIT_CURRENT_MONTH ]; then
fi
# terminazione if [ $CURRENT_YEAR -eq $REQUESTED_YEAR ]; then
fi
# Si controlla se l'anno pervenuto e' maggiore dell'anno corrente
if [ $CURRENT_YEAR -lt $REQUESTED_YEAR ]; then

    echo "<title> $FORM_month_name </title>"
    echo "<p><center>"
    #echo "<p><center><h2>"
    #echo "$TITLE</h2> <p> "
    echo "Non e' possibile procedere "

##### log_script #####

#!/bin/tcsh
#set lsfile=`/bin/ls $argv`
#set lsfile=`$argv`
#set TEMP_PREFIX="temp."
set STATIST_PREFIX="statist."
set LOG_PREFIX="logs."
set LOG_POSTFIX="total"
set POSTFIX=""
set OUTPUT_DIR="/usr/local/etc/httpd/statistics/"

```

```

set SCRIPT_DIR="/usr/local/etc/httpd/script/"
set FEED_RESULTS="feed_results"
#
if ($#argv) then
  foreach file ($argv)
    if (-s $file) then
      set POSTFIX=`basename $file | sed -e s/logs.// | awk '{printf("%s\n",
substr($0,1,3))}'`;
      cat $file >> $OUTPUT_DIR$LOG_PREFIX$LOG_POSTFIX;
      #echo " Effettuata la lettura di $file e la creazione di
      $OUTPUT_DIR$LOG_PREFIX$LOG_POSTFIX "
    endif
  end
endif
#
if (-s $OUTPUT_DIR$LOG_PREFIX$LOG_POSTFIX) then
  set BASENAME_FILE=`basename
$OUTPUT_DIR$LOG_PREFIX$LOG_POSTFIX`

if (-s $OUTPUT_DIR$STATIST_PREFIX$POSTFIX ) then
  rm $OUTPUT_DIR$STATIST_PREFIX$POSTFIX
endif

set POSTFIX=`basename $file | sed -e s/logs.// | awk '{printf("%s\n",
substr($0,1,3))}'`;

$SCRIPT_DIR$FEED_RESULTS
$OUTPUT_DIR$LOG_PREFIX$LOG_POSTFIX | awk '{printf("%s <p> \n",
$0)}' >> $OUTPUT_DIR$STATIST_PREFIX$POSTFIX;
  rm -rf $OUTPUT_DIR$LOG_PREFIX$LOG_POSTFIX
endif
#
##### feed_results #####

```

```
#!/bin/tcsh
set SCRIPT_DIR="/usr/local/etc/httpd/script/"
set CHANGE_MONTH="sed_change_month";
set AWK_USERS="awk_users"
set LIST_NEW="list_new"
#touch infor_users;
touch $SCRIPT_DIR$LIST_NEW;
awk '{if ( ($3 != "-") && ($3 != " ") ) print $3,$4}' $1 | sort -n +1 | sed -e 's/ \[/
/g' >> $SCRIPT_DIR$LIST_NEW;
sed -f $SCRIPT_DIR$CHANGE_MONTH $SCRIPT_DIR$LIST_NEW | awk -F/
'{print $1, $2, $3}' | awk -F: '{print $1, $2, $3, $4}' | awk -f
$SCRIPT_DIR$AWK_USERS
rm -rf $SCRIPT_DIR$LIST_NEW;
```

```
##### awk_user #####
```

```
BEGIN {FS = " "; x="null"; count[x]=0}
{if (($1 != x) && (x != "null")) {printf "%4d %10s %s\n", count[x], x, y/"s"/"v";
count[$1]=1; x=$1; y=$2; s=$3; v=$4 } \
else \
  if (x == "null") {count[$1]++; x=$1; y=$2; s=$3; v=$4} \
  else \
    if ($4 < v) {count[$1]++; y=$2; s=$3; v=$4} \
    else \
      if ($3 < s) {count[$1]++; y=$2; s=$3} \
      else \
        if ( ($2 < y) && ($3 == s) ) {count[$1]++; y=$2 } \
        else {count[$1]++} }
END {printf "%4d %10s %s\n", count[x], x, y/"s"/"v" }
```

Bibliografia

[Albano 85]

A. Albano e R. Orsini, "Basi di Dati", *Boringhieri*, Torino, 1985.

[Bach 88]

M. J. Bach, "The design of the UNIX operating system", *Prentice-Hall*, 1988.

[Chandler 95]

D. M. Chandler, "Running a Perfect Web Site", *MacMillan Information SuperLibrary*, 1995.

[Comer 90]

D. E. Comer, "Internet With TCP/IP", *Prentice-Hall*, 1990.

[Krol 94]

E. Krol, "The Whole Internet Users Guide and Catalog", *Prentice-Hall*, 1994

[Lemay 95]

Laura Lemay, "Il Manuale HTML", *McGraw-Hill*, 1995.

[Martin 94]

J. Martin, K. K. Chapman, J. Leben, "LAN", *Prentice Hall*, 1994.

[Stevens 90]

W. R. Stevens, "UNIX Network Programming", *Prentice Hall*, 1990.

[Tanenbaum 91]

A. S. Tanenbaum, "Computer Networks", *Prentice-Hall*, 1991.

[Zanetti 94]

G. Zanetti, "LINUX", *Libreria Progetto*, Padova, 1994.