

ASPETTI TECNOLOGICI: LE APPLICAZIONI WEB UTILIZZATE IN DRONET

Ermanno Ancona ¹⁾, Diego Perini ²⁾, Antonio Lonardi ¹⁾

1. *Web Division –Dipartimento delle Dipendenze - Azienda ULSS 20 Verona*

2. *NW BOX S.r.l.*

INTRODUZIONE

La seguente trattazione illustra in generale gli aspetti legati ad un'applicazione web soffermandosi poi, in particolare, sulle tecnologie adottate per lo sviluppo di Dronet giustificando di volta in volta le scelte fatte rispetto alle alternative disponibili.

Un'applicazione web, a livello elementare, si basa su un'architettura di tipo client-server, ossia un server centrale, che un computer ed un software client. Entrambi sono responsabili, seppure in proporzione diversa, delle elaborazioni necessarie all'esecuzione dell'applicazione stessa. Nulla impedisce, naturalmente, che la parte client e la parte server possano coesistere e lavorare su di un'unica macchina.

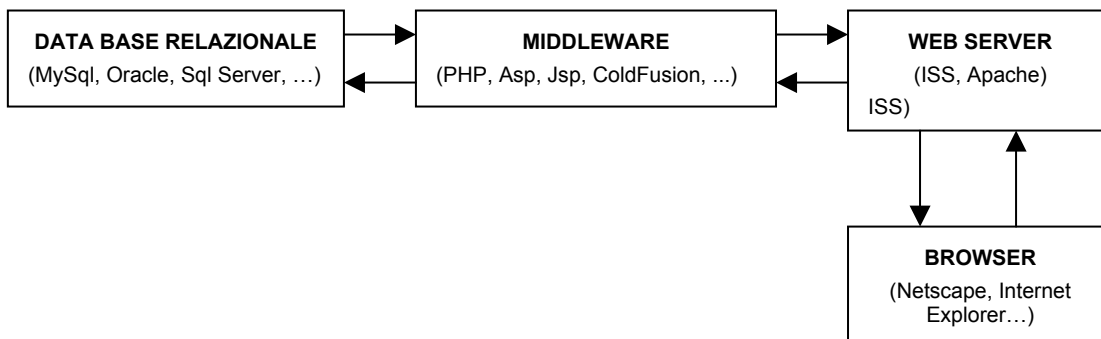
Il client da prendere in considerazione nel caso di specie è il browser web (Internet Explorer, Netscape Navigator, Mozilla, Opera etc.). La funzione principale di questi browser è interpretare i tag del linguaggio HTML, Hypertext Markup Language, che determinano come dovrà apparire la pagina web all'utilizzatore finale. Naturalmente i browser moderni non si limitano ad interpretare l'HTML, ma supportano ad esempio Javascript o XML, oltre ai più vari Plug-in: da Flash a Shockwave, a RealPlayer.

La maggior parte della potenza elaborativa viene però fornita dal server: uno specifico software, chiamato server web, si incarica di comunicare con il browser che lo interroga, un database server si incarica invece di archiviare tutti i dati che vengono messi a disposizione dall'applicazione.

Tra di essi si frappone uno strato, il cosiddetto middleware (generalmente un linguaggio di scripting), con il compito di metterli in relazione ed inoltre capace di eseguire delle elaborazioni sulle informazioni che transitano tra i due.

La figura seguente schematizza quanto brevemente descritto:

Fig. 1: Schematizzazione delle risorse tecnologiche per lo sviluppo di una applicazione web



APACHE/PHP/MYSQL

La scelta adottata per Dronet si basa sul trinomio Apache/PHP/MySQL

Il sistema che risulta dall'unione di questi tre elementi è, a nostro giudizio e a giudizio di molti, stabile e sicuro, oltre che particolarmente economico. Nessuno di questi software è a pagamento, ma scaricabile gratuitamente dai siti dei gruppi di sviluppo. Sono tutti prodotti della filosofia "open source": i sorgenti dei programmi sono liberamente distribuiti e modificabili secondo i termini della licenza GPL (General Public License). Il supporto su tali applicazioni è ottenibile in mailing list e newsgroup, oltre che tramite la documentazione disponibile on line, sia sui siti d'origine, che su migliaia di altri siti nelle più disparate lingue. Va sottolineato che la scelta iniziale di utilizzare un sistema operativo Unix-like, le cui ragioni sarebbe troppo lungo discutere, ovvero Linux, ha influenzato l'adozione di questi software. Apache, PHP e MySQL, anche se disponibili per tutte le principali piattaforme, sono stati progettati e sviluppati per sistemi Unix-like. A tutt'oggi sono la combinazione maggiormente utilizzata dai piccoli-medi server su internet.

Fig. 2: Logo della software House Apache



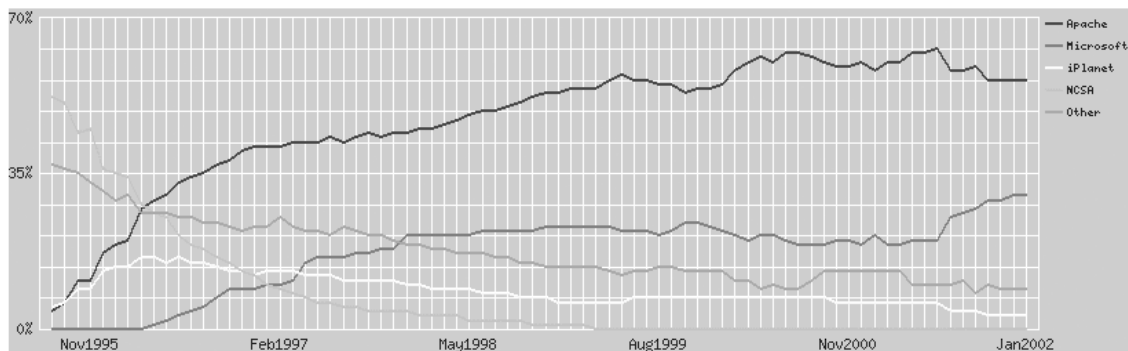
Come già accennato il server web svolge un compito apparentemente semplice: attende le richieste dei client dal web e fornisce di conseguenza le pagine. In realtà le cose sono più complesse e la stabilità di questa applicazione è fondamentale per il funzionamento di tutto il sistema. I web server disponibili oltre ad Apache sono molti, sia commerciali che open source, tra questi ricordiamo Microsoft IIS (Internet Information Server), Lotus Domino, Netscape Enterprise, Xitami, Sambar Server, Savant, Microsoft PWS. Informazioni a riguardo sono disponibili alla url: <http://serverwatch.internet.com/webservers.html>.

Due dominano il panorama e, per così dire, si spartiscono il web: IIS e Apache.

Internet Information Server è strettamente legato a windows come piattaforma e ad Asp (Active Server Pages) di Microsoft come middleware. Negli ultimi tempi IIS ha mostrato una certa flessione nel suo utilizzo. I motivi principali possono essere due: in primo luogo i buchi di sicurezza troppo spesso evidenziati, in secondo la maturità raggiunta dal suo rivale anche nella versione per windows.

Le statistiche riportate da netcraft (<http://www.netcraft.com/survey/>) vedono Apache utilizzato sul 60% del web contro un 19% del rivale IIS.

Fig. 3: Market Share for Top Servers Across All Domains August 1995 - January 2002



Il Server Apache nasce dal più "anziano" NCSA (come del resto IIS) con patch di terze parti: secondo alcuni il nome apache deriverebbe proprio da queste patch. Come già accennato fa parte della corrente di pensiero open source, che ha come pilastro filosofico rendere di dominio pubblico il codice sorgente dei propri applicativi. Questo permette a chiunque ne abbia voglia e necessità di modificare e personalizzare il programma a proprio piacimento. Grazie a questo modo di intendere il software migliaia di programmatori in tutto il mondo hanno partecipato e tuttora partecipano allo sviluppo di Apache facendolo evolvere a velocità prodigiosa.

Quali sono le ragioni che portano a scegliere Apache, oltre la sua diffusione sul web ed il suo essere open source e gratuito? Principalmente le sue performance, la sua velocità e robustezza oltre all'elevatissimo numero di moduli aggiuntivi che integrano le capacità del server. PHP stesso viene spesso eseguito come estensione ovvero modulo di apache.

Inoltre Apache è multi piattaforma, cioè disponibile per un grande numero di sistemi operativi, basati su architetture hardware completamente differenti (da Microsoft Windows, a Solaris di Sun Microsystem). È possibile creare delle applicazioni locali usando apache/php/mysql, magari sotto Windows 98, e spostarle su un server Unix senza quasi apportare modifiche. Ovviamente le prestazioni sono molto legate all'hardware ed al sistema operativo utilizzato. Tipicamente è preferibile utilizzare un sistema operativo della famiglia Unix su server con grossi carichi, per ragioni di sicurezza e stabilità.

La nuova versione di Apache, la 2.0, tra breve disponibile, preannuncia modifiche che aumenteranno performance e flessibilità. La struttura modulare risulterà ancora più spinta dando sicuramente luogo ad un'esplosione di nuovi moduli. Il modulo MPM Multi Processing Modules migliorerà il multithreading permettendo al server di adattarsi in modo ancora più efficiente al carico di lavoro momentaneo. L'APR, Apache Portable Run Time, migliorerà le prestazioni delle piattaforme non Unix e permetterà l'utilizzo di API native realizzando l'indipendenza dalle emulazioni POSIX.

Per ultima va menzionata l'integrazione del supporto IPV6, già comunque distribuita come patch nel passato.

Per taluni un difetto attribuibile a questo web server è la scarsità di tools grafici per la sua amministrazione. Invero molti preferiscono, come noi, editare a mano i file di configurazione: scrivere direttamente nell'htpd.conf permette un maggiore controllo dell'applicazione specie se si vogliono utilizzare nuovi moduli.

PHP

Fanno parte del cosiddetto middleware, tutti quei linguaggi di programmazione che lavorano a stretto contatto con il web server per interpretare le richieste provenienti da internet, elaborarle, interagire con il database ed infine soddisfarle. Semplificando possiamo dire che il linguaggio di scripting dice al web server cosa mostrare al browser ed è qui che viene svolta la maggior parte del lavoro.

Nell'evoluzione di Internet il passaggio da un web statico ad uno dinamico, ossia in grado di interagire in qualche modo con il client, si realizzò inizialmente tramite linguaggi di scripting "client side". Il codice di tali linguaggi, tra cui sicuramente JavaScript è il principale, viene inserito nella pagina html ed eseguito direttamente dal client.

Ben presto si sentì l'esigenza di qualcosa di più, di uno strumento in grado di generare pagine al momento della richiesta del client anche in funzione delle informazioni fornite dal browser. Ci voleva qualcosa che fosse indipendente dall'ambiente dell'utente, sia da impostazioni e preferenze, sia dal browser, sia dal sistema operativo. Era necessario creare una dinamicità gestita dal server o come si suol dire server side, espressione tradotta in italiano come "lato server".

Inizialmente la soluzione adottata fu di utilizzare programmi residenti sul server e che, tramite le Common Gateway Interface, comunicassero con il web server, i cosiddetti CGI. Ancora oggi vengono comunque largamente utilizzati, soprattutto programmi scritti in PERL, Practical Extracting and Reporting Language. Uno dei motivi che portò alla ricerca di altre tecnologie rispetto alle CGI è la mancanza di scalabilità che le caratterizza: ad ogni richiesta corrisponde l'esecuzione di una nuova istanza del programma con conseguente carico sulle risorse di sistema.

Ben presto si impose un modello nuovo per la dinamicità lato server, basato su linguaggi immersi nell'HTML o HTML-embedded. Il codice viene inserito tra appositi marcatori ed eseguito sul server prima di fornire la pagina. Risulta completamente invisibile al client, che vede solo quanto generato dall'esecuzione dello script. In pratica il file, individuato da una particolare estensione nota al server web, viene filtrato dall'interprete del linguaggio prima di restituire l'output al client. L'interprete lascia passare inalterati tutti i tag HTML ed esegue soltanto le istruzioni incapsulate tra due speciali marcatori.

Attualmente le principali e più comuni tecnologie lato server sono:

- Common Gateway Interface (CGI).
- Application Program Interface (API) proprietarie.
- Microsoft Active Server Pages (ASP).
- Java Server Pages (JSP).
- Cold Fusion Markup Language (CFML).
- Personal Home Page (PHP)

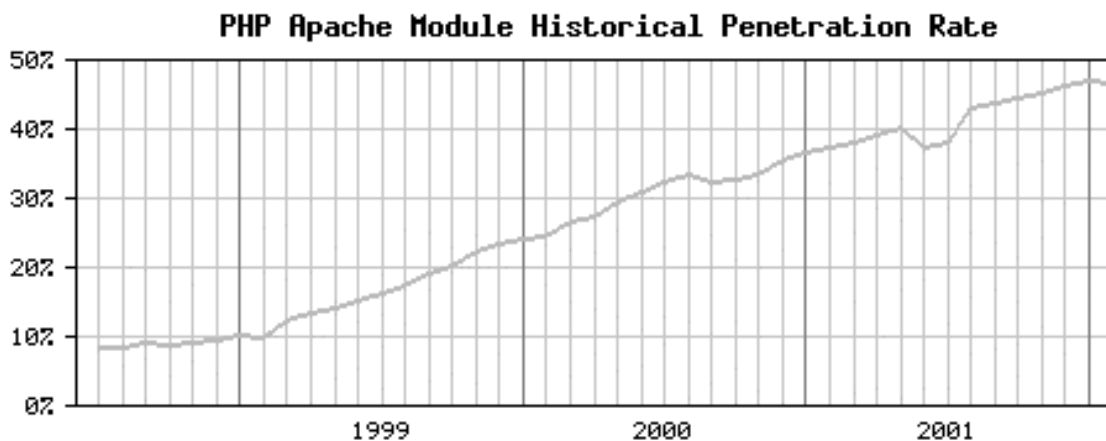
PHP (Personal Home Page) è un linguaggio di programmazione interpretato, server side, creato nel 1994 da Rasmus Lerdorf. La svolta più importante nella vita di PHP si ebbe quando Andi Gutmans e Zeev Suraski, software engineers, avendo trovato casualmente un bug, decisero di unirsi al gruppo di sviluppo per migliorarne il codice. A loro si deve il motore di scripting Zend (www.zend.com) che migliorò notevolmente le prestazioni del linguaggio nelle versioni 3 e 4. Zend è proprietà legale di Zend Technologies Ltd, ma viene concesso in licenza d'uso gratuita secondo i termini della licenza QPL.

Fig. 4: Loghi di PHP e Zend

Il successo sulle pagine dei media di PHP è certamente inferiore a quello ad esempio di Linux o Apache, malgrado ciò, secondo alcune statistiche, gira oggi su più di un milione host.

Di seguito si riportano i risultati di un'inchiesta di E-Soft che mostrano il trend di utilizzo di PHP come modulo di Apache:

(http://www.securityspace.com/s_survey/data/man.200201/apachemods.html?mod=UehQ)

Fig. 5: Trend di utilizzo di PHP come modulo Apache

Naturalmente la penetrazione di PHP nel web non è di per sé un motivo sufficiente per scegliere di utilizzarlo per le proprie applicazioni. Allora perché scegliere PHP?

Premesso che spesso sui linguaggi di programmazione si scatenano guerre di religione e diatribe infinite e senza voler entrare in contrapposizione con qualunque altro strumento server side, esporremo alcune considerazioni che, a nostro personale giudizio, possono portare a prediligerlo.

PHP offre un mix di potenza, velocità e facilità d'uso, è lo strumento che permette di sviluppare più rapidamente applicativi web. È evidente che un programma scritto in C risulti più veloce, ma richiede maggiore impegno e più tempo al programmatore. Php ha una sintassi molto semplice e snella derivata in parte da altri linguaggi: C/C++, Java, Perl. È molto flessibile e, grazie alla vastissima libreria di funzioni, operazioni come interrogazioni di database o utilizzo di socket TCP, creazione dinamica di immagini (tramite la libreria GD) o di documenti PDF (tramite la libreria PDF) possono essere gestite in modo immediato.

Vi sono funzioni per leggere e inviare header HTTP, particolarmente utili quando lo si utilizza in relazione ad altri linguaggi come XML o WML, oppure per semplici redirezioni lato server o invio di cookie. Permette di interagire con altri servizi utilizzando i principali protocolli IMAP, SNMP, NNTP, POP3. Va sicuramente menzionato il meccanismo delle sessioni, nativo dalla versione 4, che risolve il problema della mancanza di stato del protocollo HTTP. Tale sistema permette di associare uno stato ad ogni client che accede ad un sito: il server memorizza le informazioni di sessione su file o database. Il client da parte sua tramite un identificativo univoco di sessione, SID, fornisce al server le informazioni necessarie per associargli la sessione corrispondente.

Probabilmente la sua caratteristica principale è il supporto ad un elevatissimo numero di DBMS, grazie a librerie specifiche o attraverso l'interfaccia ODBC, accede con facilità a MySQL, ma anche ad Oracle così come a DB2 o Informix.

Nonostante sia un linguaggio interpretato vanta prestazioni notevoli, ulteriormente migliorate dalla versione 4.

È multiplatforma e può funzionare, oltre che sul supporto nativo dei sistemi operativi Unix-like, anche sotto Windows2000/NT/98, sia con IIS che con Apache. Vi è qualche dubbio, a dire il vero, sulla sua stabilità con IIS quando operano su server soggetti a grossi carichi di lavoro. La portabilità delle applicazioni è possibile anche con Asp o Jsp, ma sicuramente non con le stesse performance e con la stessa necessità di minimi adattamenti.

È costantemente migliorato: migliaia di programmatori di talento partecipano al suo sviluppo secondo la filosofia open source; ha una struttura estremamente modulare che facilita l'integrazione continuativa di nuove funzionalità.

Visto il forte senso di comunità tipico dell'open source, ci sono sempre persone disponibili ad aiutare un programmatore su forum e mailing list. Risulta particolarmente agevole il reperimento di documentazione, oltre che di classi, librerie, script o anche interi software molto complessi già pronti e distribuiti gratuitamente.

In chiusura una ragione che potrebbe sembrare banale ed invece non lo è: perchè uso MySQL. L'integrazione con tale database è infatti perfetta e l'accesso diretto via socket rende la coppia PHP/MySQL la più veloce e performante del web.

MySQL

Un database è una collezione di dati che viene amministrata e organizzata tramite un software detto DBMS, Data Base Management System. Tale DBMS costituisce uno strato tra le applicazioni ed i dati veri e propri. Grazie a ciò l'utente e le applicazioni non accedono ai dati così come sono fisicamente memorizzati, ma ne vedono solamente una rappresentazione logica. L'indipendenza dal livello fisico permette un elevatissimo grado di astrazione, tanto che un amministratore di database, se necessario può memorizzare i dati in maniera differente per esempio partizionandoli su più dischi o cambiare il DBMS senza che le applicazioni, e quindi gli utenti, ne risentano eccessivamente. L'importante è che non venga cambiata la rappresentazione logica dei dati, lo "Schema del database", che è la forma di rappresentazione dei dati più a basso livello cui un utente può accedere. Il miglior sistema per archiviare ed interagire con informazioni complesse è il database relazionale, il modello senza dubbio più utilizzato dagli anni 80 ad oggi, ma originariamente teorizzato da E.F.Codd in un articolo del lontano 1970.

Tra i principali database relazionali in commercio citiamo Oracle, Sql Server, Sybase, Informix, Db2, mentre tra quelli open source MySQL, Postgres, Interbase recentemente rilasciato da Borland. Tutti implementano lo standard SQL, Structured Query Language, un linguaggio di interrogazione strutturato che permette di interagire agevolmente con insiemi di dati.

Fig. 6: Logo MySQL

Lo svantaggio principale è forse l'implementazione di un subset di SQL limitato, in particolare non supporta right join, select di select, integrità referenziale, transazioni.

Molte di queste funzioni sono comunque, con un po' di esperienza, "aggirabili" senza eccessivi sforzi. Inoltre, come tutti i prodotti open source, viene costantemente migliorato e tra le successive aggiunte sono previsti il supporto alle transazioni e alle procedure archiviate (stored procedure).

A livello di applicativi, si fa sentire la carenza di un'interfaccia grafica potente pur esistendo prodotti per la sua amministrazione come PhpMyAdmin o MySQLFront. D'altro canto questo obbliga ad un controllo preciso del codice SQL e a una maggiore padronanza e familiarità nella programmazione.

La definizione che di MySQL da il team di sviluppo nella documentazione ufficiale è: "MySQL is a very fast, multi-threaded, multi-user and robust SQL (Structured Query Language) database server". In effetti è un database SQL server molto veloce, multi-processo, multi-utente e robusto. Può quindi rispondere a diverse richieste contemporaneamente, può essere utilizzato contemporaneamente da diversi utenti, ma soprattutto è in grado di gestire grandi quantità di dati con risorse hardware limitate.

MySQL è stato sviluppato per poter gestire database con centinaia di migliaia di tabelle: il suo team di sviluppo lo sta utilizzando dal 1996 su un database con 10,000 tabelle, di cui più di 500 hanno più di 7 milioni di record, per un totale di circa 100 Gigabyte di spazio occupato.

Scritto in C e C++, è stato testato con numerosi compilatori, supporta fino a 50,000,000 di record e differenti set di caratteri, inclusi ISO-8859-1 (Latin1), big5 (Cinese), Scandinavo ed altri. L'ordinamento è basato sul set di caratteri scelto, il server può restituire i messaggi di errore in differenti lingue.

MySQL risulta essere completamente multiprocesso, è compatibile con differenti piattaforme, permette una gestione flessibile e sicura di password e privilegi, controlli basati sull'host oltre a connessioni crittografate. È infatti possibile permettere l'accesso al database unicamente al localhost garantendo interrogazioni solo da pagine che girano sul proprio web server. Supporta inoltre le funzioni ODBC 2.5.

È vero che Oracle, Sybase, Informix hanno database molto potenti, ma i costi per tali prodotti sono obiettivamente alti, mentre MySQL è completamente gratuito e per database di piccole e medie dimensioni è sicuramente il più veloce sul web.

Infine è d'obbligo ribadire il già citato rapporto idilliaco che lo lega al linguaggio PHP, con il quale costituisce una formidabile accoppiata per lo sviluppo di siti web.

CONCLUSIONI

Nella realizzazione di un'applicazione web non esistono regole generali che si adattano a tutte le circostanze, esistono però alcune variabili che è necessario valutare con attenzione prima di operare delle scelte.

Il codice disponibile. In molti casi il problema che vogliamo risolvere non è così diverso da problemi che già altri hanno brillantemente risolto. La possibilità di riutilizzare codice già scritto è un ottimo motivo per preferire un linguaggio di programmazione ad un altro. Probabilmente, da questo punto di vista, è molto più facile realizzare un forum in PHP/MySQL piuttosto che in ASP/Access.

Le prestazioni. In alcuni casi è necessario valutare attentamente le prestazioni che una piattaforma è in grado di offrire. Tipicamente, un sito di e-commerce ambizioso non può utilizzare Access. Allo stesso modo, se si necessita di utilizzare funzioni come l'integrità referenziale, non è consigliabile utilizzare MySQL.

Il grado di conoscenza della piattaforma di sviluppo. Non è un fattore da sottovalutare: per un progetto semplice si può scegliere una qualunque piattaforma con cui si abbia una qualche familiarità, non è così per progetti avanzati e impegnativi, o per progetti che richiedano un rapido sviluppo.

Le risorse hardware e software. In alcuni casi ci si trova di fronte a dati di fatto che rappresentano dei forti vincoli e di cui si deve tenere conto. Per quanto riguarda l'hardware, i vincoli principali sono di "disponibilità" o di prestazioni: non è possibile installare IIS su un vecchio Pentium. Analogamente dal punto di vista software la disponibilità o la necessità di alcuni tool di sviluppo può consigliare una determinata scelta. Ad esempio i principali strumenti di programmazione visuale sono per ASP o JSP.

Per PHP non esistono, invece, tool visuali particolarmente utili e soprattutto efficienti per velocizzare la programmazione: risulta dunque una scelta consigliabile per chi ha familiarità con la programmazione ed è disposto a scrivere la totalità del suo codice a mano. Tenendo presente che sempre e solo in questo modo si otterrà il controllo completo di un'applicazione.

Le risorse economiche. In conclusione non si può dimenticare che alcune piattaforme hanno un costo non indifferente, sia per quanto riguarda le licenze, sia per quanto riguarda il supporto, l'assistenza e gli upgrade.

BIBLIOGRAFIA

- DuBois P., MySQL, Edizioni New Riders
Gibelli L, Apache 2.0 Beta novità in vista. Linux&c anno3 num17 pgg36-38
Greenspan J. Brad Bulger, Sviluppare applicazioni per database con mysql/php, Edizioni Apogeo
Kabir M.J., Apache Server - guida per l'amministratore, Edizioni Jackson Libri-IDG Books
Ratschiller T., Gerkeen T PHP 4.0 Applicazioni Web. Edizione Addison Wesley
<http://www.apache.org>
<http://www.mysql.com>
<http://www.php.net>
<http://phpwizard.net>
<http://www.phpcenter.it/>
<http://freephp.html.it/>
<http://www.latoserver.it>
<http://serverwatch.internet.com/webservers.html>
<http://www.netcraft.com>
<http://www.html.it>
<http://www.manucorp.com/>
<http://www.subnet.it/>
<http://www.trovaguide.it>
<http://www.baravalle.it>