A motore aperto si può scegliere meglio

Il primo passo per identificare quello più adatto è conoscere come funzionano.

Excite forse è il migliore ed è gratis

di Marco Iannacone <ianna@iol.it>

Una delle funzioni più utili per la quale un computer può essere utilizzato all'interno di un'azienda è la possibilità di effettuare delle ricerche all'interno di centinaia di documenti recuperando quelli che soddisfano particolari requisiti. Non mi riferisco alla ricerca di informazioni presenti all'interno di una base di dati strutturata quale quella di un database relazionale, ma in quell'insieme eterogeneo di documenti di formato elettronico che viene creato all'interno di un'azienda.

Il grande problema di questa mole di informazioni è la difficoltà intrinseca che comportano nel reperimento di dati precisi. Quando in gioco ci sono pagine e pagine, la ricerca per parole chiave si rivela inutile, perchè è molto facile che l'elemento oggetto della ricerca per quella particolare query non venga utilizzato come parola chiave. L'ideale sarebbe poter effettuare delle ricerche all'interno dei documenti per poter verificare la presenza della parola incriminata.

Ancora una volta questo si dimostra uno di quei casi per cui la presenza di una macchina Unix all'interno di un'azienda si può rivelare utile per risolvere il problema (noi prenderemo in esame l'uso di Linux che assicura costi molto più ridotti). Il problema, infatti è stato già risolto diversi anni fa, anche se oggi, nuovi strumenti si affacciano sul mercato rendendo questo tipo di ricerca ancora più potente e versatile. In questo articolo ci proponiamo di analizzare le tecnologie già consolidate e di vedere quali innovazioni apportano i nuovi prodotti.

Con la giusta tecnologia è possibile consentire ad utenti remoti di effettuare ricerche in documenti contenuti in uno o più server sulla vostra rete: è possibile limitare la ricerca ad un particolare sottoalbero delle vostre directory, oppure effettuare un'analisi dell'intero disco.

Ovviamente se si vuole che i documenti siano visualizzabili attraverso un browser, è necessario convertirli preventivamente in formato HTML, oppure utilizzare appositi plug-in che consentano la visualizzazione di quel formato (per esempio Acrobat Reader per il formato PDF). Va tenuto presente, tuttavia, che la maggior parte dei motori di tipo gratuito consente di indicizzare solo documenti testuali o HTML.

Sono molti i pacchetti (distribuiti con la licenza GNU) che consentono di implementare questi tipi di servizi. I più noti sono quelli basati su *freeWAIS*, un'implementazione distribuita gratuitamente del potente protocollo WAIS (Wide Area Information Server) -sviluppato originariamente da *Thinking Machine Corporation*. In aggiunta ai motori basati su WAIS ne esiste un'altra serie - non meno potente - basata su Glimpse.

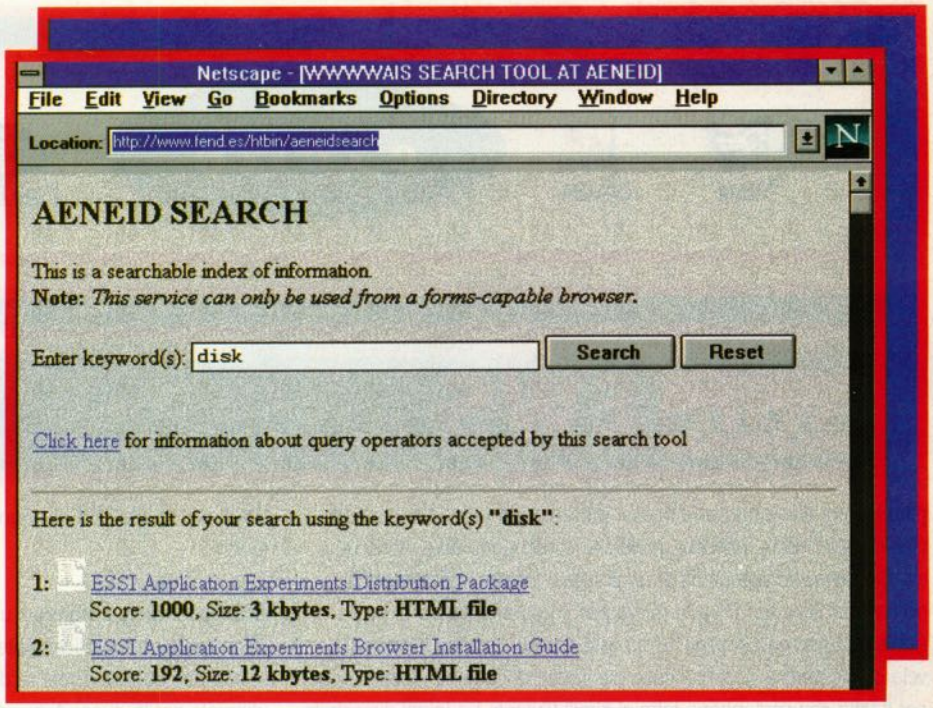
Entrambi i tipi di motori comprendono solo le funzionalità di indicizzazione di un gruppo di documenti e la capacità di effettuare delle ricerche all'interno di questi indici in maniera molto veloce, fornendo il nome dei documenti che soddisfano la richiesta assieme ad un valore che rappresenta quanto il documento soddisfi i criteri impostati. Per poterli utilizzare attraverso un webserver (come i più noti motori di ricerca) è necessario installare uno o più script di gateway. I gateway più noti (anch'essi gratuiti) sono **webglimpse** (per Glimpse) e **wwwwais** per il mondo derivato da freeWAIS. Sicuramente il motore più semplice da configurare e quello più adatto ad indicizzare pagine HTML è

Marco Iannacone, si occupa di UNIX, networking e Internet come sistemista e consulente. Lavora come System Administrator e responsabile dei servizi internet presso la Dun & Bradstreet Kosmos

SWISH (è creato apposta per dar maggior peso a parole trovate nel titolo rispetto a quelle contenute nel documento e ad ignorare le parole contenute all'interno dei tag HTML). SWISH (Simply Web Indexing System for Humans), creato da Kevin Hughes riesce inoltre a creare indici che hanno dimensioni inferiori rispetto a quelli generati da freeWAIS (tenete presente, tuttavia, che la loro dimensione è all'incirca il 50% della somma dei documenti che si desidera indicizzare). E' possibile scaricare SWISH da: **ftp://ftp.eit.com/pub/web.software/swish/** mentre la documentazione è disponibile presso: **http://www.eit.com/software/swish/swish.html**
Una volta compilato e installato il programma, è sufficiente editare il file di testo swish.config nella directory conf per indicare la directory da indicizzare, quale tipo di file e in quale directory creare l'indice. A questo punto lanciando il programma (**swish -c nome_file_di_configurazione**), verrà creato l'indice che sarà poi usato per le ricerche. Si possono ovviamente creare indici differenti per zone diverse del disco, per tipo di file, etc.
Come dicevamo, le ricerche possono essere effettuate da linea di comando, oppure utilizzando un gateway web come wwwwais. Potete scaricare wwwwais da **ftp://ftp.eit.com/pub/web.software/wwwwais/** e prima di compilarlo, modificate il file wwwwais.c in modo che l'istruzione CONFILE indichi l'esatto percorso di dove avete intenzione di mettere il file di configurazione (wwwwais.conf). La configurazione di wwwwais è molto semplice e la documentazione contiene anche una pagina web di esempio per poter iniziare a effettuare ricerche una volta installato wwwwais nella directory cgi-bin del proprio web server. L'unica aggiunta che mi sento di fare all'abbondante documentazione disponibile in rete è quella di utilizzare congiuntamente a wwwwais anche il programma **print_hit_bold.pl** (scaricabile da http://ewshp2.cso.uiuc.edu/print_hit_bold.pl) in modo che le pagine che soddisfano la ricerca presentino in grassetto al loro interno le parole cercate. Per abilitare questa funzione è sufficiente aggiungere le seguenti istruzioni nel file wwwwais.conf:

```
SourceRules prepend /cgi-bin/print_hit_bold.pl
SourceRules append ?$KEYWORDS#first_hit
```

I motori di ricerca di cui ho parlato fino ad ora (swish. freeWAIS e in parte anche glimpse), pur essendo degli ottimi prodotti che svolgono dignitosamente il loro lavoro e che probabilmente soddisfano le esigenze della maggior parte delle piccole/medie aziende hanno tuttavia dei chiari limiti. Non esiste infatti alcuna *intelligenza* dietro questi motori: si limitano a reperire le informazioni ricercate all'interno di indici costruiti in maniera molto empirica. Reperire l'informazione voluta all'interno di una moltitudine di documenti, non è tuttavia così semplice, come sanno bene coloro che utilizzano la rete regolarmente per trovare le informazioni che gli interessano.. Chi quindi, meglio delle società che hanno messo a punto i famosi *ragni* (quali yahoo, lycos, altavista, excite) poteva disporre di una tecnologia *intelligente* per reperire le informazioni? E infatti ormai da più di un anno i principali motori di ricerca hanno rilasciato dei prodotti (spesso distribuiti gratuitamente) che consentono di effettuare ricerche all'interno di uno o più pc utilizzando la stessa logica usata sulla Rete. Excite offre a mio avviso il prodotto più interessante, EWS (Excite for Web Servers) - basato sulla tecnologia ICE (Intelligent Concept Extraction) - rilasciandolo gratuitamente anche per la piattaforma Linux. Ci sono due aspetti da considerare quando si vuole valutare la bontà di un motore di ricerca: *recall* e *precision*. Con recall si intende la percentuale di tutti i documenti rilevanti contenuti all'interno del database che il motore riesce a recuperare. Se un database contiene 100 documenti che soddisfano una determinata query e ne vengono recuperati solo 40 il fattore recall viene considerato 0,40. Precision, invece, indica la percentuale di documenti rilevanti rispetto al totale dei documenti recuperati. Se per esempio il risultato di una ricerca recupera 50 documenti, ma solo 40 sono rilevanti, il fattore precision avrà valore 0,80. Ovviamente sia il pescaggio (recall) che la precisione (precision) sono importanti e in un mondo ideale dovrebbero essere entrambi uno, ma ciò non accade mai nella realtà. Com'è facilmente intuibile, nel valutare la bontà di un sistema è importante tener presente entrambi gli aspetti. Le tecnologie che si sono succedute nel tentativo di migliorare i risultati delle ricerche sono diverse: cerchiamo di capire le caratteristiche di ognuna di esse.



Le ricerche **booleane** sono le prime ad essere state adottate: consentono all'utente di specificare le parole con cui effettuare le ricerche collegandole tra loro con gli operatori logici AND e OR. Pur essendo molto semplice questo metodo non da risultati interessanti. Si ottiene un alto pescaggio (recall) ma una precisione (precision) piuttosto bassa.

Le ricerche **fuzzy boolean** consentono di specificare una serie di

parole e poi effettuano una ricerca presentando prima i documenti che le contengono tutte, poi quelli che contengono tutte meno una, etc. Ciò consente agli utenti di aggiungere termini ai parametri di ricerca, mantenendo un alto pescaggio, ma aumentando la precisione, quantomeno ai vertici della lista. Questo metodo tuttavia si rivela inutile perchè non ha senso dichiarare documenti rilevanti solo per il fatto che contengono termini simili (l'utente ha infatti dimostrato di aggiungere sinonimi nel desiderio di trovare documenti rilevanti).

Le ricerche **vector based** fanno un passo avanti considerando la percentuale di ricorrenza dei termini specificati, all'interno di ogni documento, confrontandola con la ricorrenza all'interno dell'intero database. Questo sistema assume che i termini della query che sono rari all'interno dell'intero database siano più importanti e quindi fornisce una maggiore priorità ai documenti che li contengono.
Questa tecnologia utilizza un modello geometrico per creare un differente asse per ogni termine e fornisce su di esso un valore proporzionale al numero di ricorrenze all'interno di ogni documento. Ogni documento è quindi identificato da un vettore all'interno dello spazio del modello. Ciò che succede è che documenti sullo stesso argomento vengono a trovarsi nella stessa zona dello spazio ed è quindi possibile effettuare query che richiamino tutti i documenti di una determinata area. In definitiva mentre il modello basato su vettori può aumentare la precisione, non aumenterà il pescaggio perchè non sarà in grado di recuperare documenti che sono rilevanti ma che non contengono i termini impostati nella query. C'e' quindi un miglioramento rispetto alle ricerche booleane o fuzzy, ma solo in termini di precisione.

Le ricerche **Thesaurus based** cercano di migliorare il pescaggio delle vector based. In questo caso qualcuno crea manualmente un dizionario di termini che sono sinonimi di numerosi termini contenuti nel database. Quando viene impostata una query il motore ricerca anche i sinonimi aumentando in questo modo sia il pescaggio che talvolta la precisione. Sfortunatamente, a causa dell'enorme lavoro necessario a creare il dizionario, questa tecnologia è adatta a piccoli database orientati ad un argomento specifico che si conosce a fondo.

Automatic Query expansion usa anch'esso un dizionario, ma questo viene creato automaticamente includendo termini che sono presenti in maniera considerevole in documenti che contengono le parole utilizzate per la query. A questo punto il motore esegue una seconda ricerca utilizzando il dizionario specifico che ha appena creato. Questo metodo fornisce miglioramenti in termini di pescaggio, ma la precisione tende ad essere modificata negativamente.

Structured Query Search è una tecnica molto sofisticata che consente di recuperare informazioni di alto pescaggio e precisione, ma richiede all'utente un tempo molto elevato per impostare una query (circa 30 minuti).

Latent Semantic Indexing search è l'unica tecnologia in questa lista che bypassa il processo Booleano. Come la ricerca vettoriale, crea una rappresentazione geometrica di ogni documento contenuto nel database. Riducendo poi queste informazioni ad una matrice ed usando manipolazioni di tipo matematico, questo metodo cerca di stabilire una relazione con termini correlati e quindi riconosce il contenuto separabile di ogni documento. Ciò migliora sia pescaggio che precisione ma la grossa mole di calcoli necessari a realizzare questo tipo di elaborazioni ne rende difficile l'uso con database di grosse dimensioni.

La tecnologia proprietaria usata da Excite e chiamata Central Tehnology è un'evoluzione del metodo Latent Semantic che richiede capacità computazionali notevolmente ridotte e dunque consente di ottenere risposte rapide. Utilizzando algoritmi statistici avanzati è in grado di catalogare i documenti in base alle correlazioni dei concetti contenuti in esso e nello stesso tempo per parole chiave. Questo metodo consente un notevole miglioramento sia in termini di pescaggio che in termini di procisione rispetto a tutti i metodi analizzati fino ad ora. Un'altra capacità di questo sistema (che deriva dal modo stesso con cui classifica i documenti) è la capacità di esaminare ogni singolo documento e di estrarre le frasi più significative per quanto riguarda quel particolare concetto fornendo all'utente una sorta di riassunto. Per questo motivo, e per il fatto che il prodotto viene distribuito gratuitamente per numerose piattaforme (tra cui Linux) mi sento di consigliare EWS di Excite come un strumento ideale per l'indicizzazione e la ricerca di testo all'interno di una azienda.

Per maggiori Informazioni:

http://www.eit.com/software/swish/swish.html
http://www.excite.com